DesignWorks™ Simulator 3.1 - Release and Installation Notes

These notes provide supplementary information on DesignWorks Simulator version 3.1, released June 10, 1993.

Thank-you

Thank-you for purchasing the DesignWorks Simulator package from Capilano Computing. We are sure that you will find the package productive and enjoyable to use. Should you have any problems, questions, or comments, please contact us at the number given below.

What you have received

This package contains the 3.1 release of DesignWorks Simulator, and will include the following:

1 Release Notes
5 Diskettes
1 Reference Manual Package
1 License Agreement
1 Serial Number label or sheet §
1 Registration Card *
1 Support Plan Enrollment Card *

* These items will not be included in upgrade packages.

§ If you purchased this package together with the Schematic package, there will be only one Serial Number sheet for both.

©1993 Capilano Computing Systems Ltd. All rights reserved.

Note re Version Numbers: In versions of DesignWorks prior to 3.0, the simulator was an integrated function of the package and not available as a separate product. Version 3.1 is actually the first release of the Simulator as a separate product and is so numbered to indicate compatibility with DesignWorks Schematic 3.1.

Installation Notes

See also: Chapter II - Getting Started in the DesignWorks Simulator reference manual.

In order to simplify installation and reduce the number of diskettes required to distribute DesignWorks, the release files are compressed onto a small number of diskettes using the Apple Installer and the StuffIt™ utility.

The Installer allows you to install a complete basic system in one operation, or to choose specific groups of items to be installed.

IMPORTANT: Schematic Version Compatibility

The DesignWorks Simulator requires DesignWorks/Schematic version 3.1.1 or later for correct operation. If you are adding the Simulator to an existing DesignWorks installation, you should install any required Schematic upgrade first.

Installation Options

You may choose to select only a subset of the files included with the Simulator if they are not required for you application. The Installer's "Customize" option allows you to select from among the following groups of items.

Note: Installing the Simulation Tools and Demo Files, etc. items include everything needed for basic Simulator operation, including working through the tutorials.

Simulation Tools	This includes the actual program modules that comprise the Simulator package. They will normally be placed in the Tools folder in your DesignWorks installation. The Simulator is not a stand-alone program and operates only in conjunction with the main DesignWorks package. These items require a total of about 600K of disk space.
Demo Files, etc.	This group includes a number of demonstration designs for use with the Simulator, a basic set of models for use with the tutorials and some utility items. These items require a total of about 1MB of disk space.
74XX Models	This group includes a large number of "external model" files which define the simulation functions for the 74XX and 54XX logic families. These items require a total of about 9 MB of disk space.
4000 Models	This group includes the models for the 4000-series CMOS devices. These items require a total of about 1.2 MB of disk space.

Optional Simulation Tools

The Simulator is made up of a number of different program modules, or "tools" that operate within the DesignWorks environment. Some of these tools are optional and can be removed after installation if they are not required. The following table summarizes the Simulator tools.

Sim	This is the primary digital simulation engine. It is required for all simulation functions.
Timing	The timing diagram display and editing tool. It does not have to be included if timing waveform display is not required.
TestPanel	The test vector entry and execution tool. It does not need to be installed if its services are not required.
SimLoad	This tool associates internal simulation models with parts as they are used and is optional if not needed.

PromPLA	This tool performs data entry and simulation functions for PROM and PLA devices. IMPORTANT: This should be installed even if you are not directly using PROM or PLA devices as many of the simulation models supplied with DesignWorks do use them internally.
RAM	This tool performs data entry and simulation functions for RAM devices.
FromABEL	This tool creates simulation models for devices described using MacABEL or other external means. It can be discarded if you will not be creating PLD or PLA devices.

The Installer places all of the above tools in the Tools folder of your DesignWorks/Schematic installation. Any tools that are not being actively used can be moved temporarily to another folder and they will not be loaded.

Program Installation Procedure

Before starting the installation procedure you should take the following preparatory steps:

1) The Installer requires that it be the only program running on the machine it is installing onto. Before starting the Installer, shut down all other applications.

> Note: If you are using Macintosh System Software version 7.0 or later, the installation will be significantly faster and more problem-free if you restart you machine without extensions. This is done by selecting the Restart item in the Special menu and holding the Shift key on the keyboard depressed until the message "Extensions Off" appears. This disables all file sharing, virus detection and other background processes that can interfere with the installer.

2) If you have any virus detection software on your machine, you may find some interaction with the Installer. It is best to disable any virus software before proceeding.

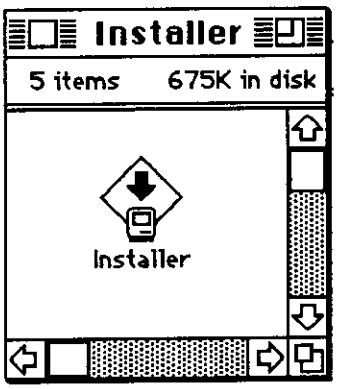
3) Depending on which disk you are installing on, the Installer may require that you disable File Sharing, or at least ensure that no folders are exported from your machine during the installation.

4) The standard installation of DesignWorks and its default libraries requires about 16 megabytes of disk space. Most of this space is taken up by parts

libraries. If you want to install the default system, you will need to ensure that this amount of space is available. You can use the "Customize" option to choose only a basic program installation to get started, or to install additional libraries. In any case, a minimal installation requires at least 5 megabytes.

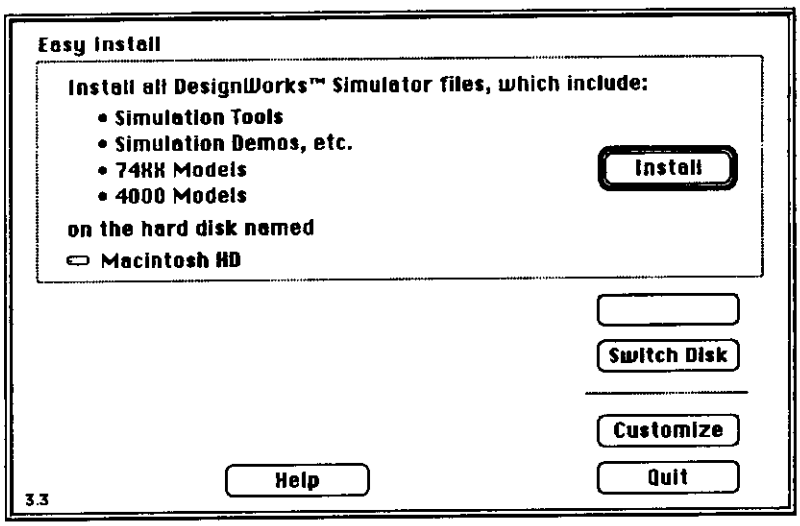
To start the installation:

1) Insert the diskette labeled "Installer". You should see the following files on the diskette:



2) Do not copy any of these files onto your hard disk! Double-click on the file "Installer". This will start up the installer program.

3) The following box will be displayed:



At this point you can choose one of two paths:

- Clicking the Install button will install all the DesignWorks Simulator files on your disk.
- Clicking the Customize button will allow you to choose which items you install.

5) The installer will now copy the files from the installation disks, and then unpack the files onto your disk. It will ask for the diskettes it requires as it proceeds.

6) The next time you start the DesignWorks program, the Simulator tools will appear in the Tools menu.

Serial Number Entry

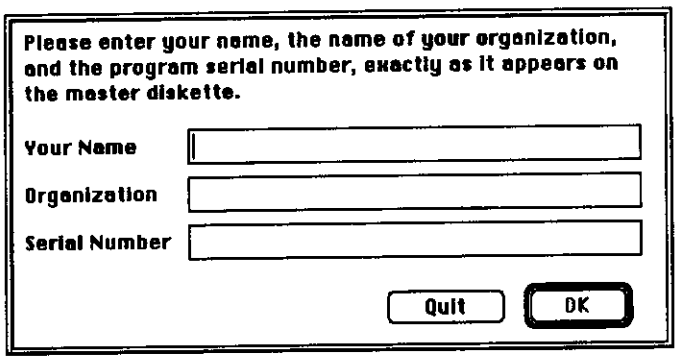
If the Simulator was purchased separately from the original DesignWorks package, you may have to enter the new Serial Number provided in order to enable the Simulator with your copy of DesignWorks. If you purchased the Schematic and Simulator packages together, then simply follow the serial number entry procedure in the DesignWorks/Schematic Release & Installation Notes.

Entering a New Serial Number

To enter a new Serial Number into an existing DesignWorks installation:

- Quit the DesignWorks program if it is already running.
- Start the program by double-clicking on the DesignWorks icon while holding the Option key on the keyboard pressed.

You will see the following serial number entry box:


Please enter your name, the name of your organization, and the program serial number, exactly as it appears on the master diskette.
Your Name
Organization
Serial Number
Quit
OK

Enter the following information:

- Your name and the name of your organization. This information will be displayed each time the program is started.
- The serial number as provided with your master diskette. This is a sequence of 20 letters and numbers. It can be entered either with or without separating spaces. All letters must be upper case.

After this information is entered once, it is kept with the program and you will not need to enter it again.

DesignWorks uses a network copy protection scheme that limits the number of users of a single copy on a network. The serial number contains encoded in it the number of authorized users and validation codes for add-on tools. You can upgrade your copy at any time by contacting Capilano Computing at the phone number given elsewhere in this manual.

Site or Network Licensing

When you purchase a copy of our software you are licensed to store and use it on a single computer at one time, making copies only for backup purposes. We have very attractive multiple unit licensing arrangements for organizations that are interested in using our software on a number of machines on the same network. Please contact us directly for more information.

Copy Protection

For your convenience, the DesignWorks disks are not copy protected in any way, allowing you to copy the program onto your hard disk or backup floppies as desired. This does **not** imply that you are allowed to make copies for use on more than one computer, in fact you are specifically prohibited from doing so by your license agreement. By respecting that agreement, you will help us continue to bring you useful software products at reasonable prices.

Technical Support

If you have any questions, problems, suggestions, congratulations or whatever, you can contact us using any of the following methods:

Phone	(604) 522-6200 (9 - 5 Pacific time, please ask for DesignWorks Technical Support)
Fax	(604) 522-3972
AppleLink	D0816,
InterNet	D0816@applelink.apple.com
Mail	Technical Support, Capilano Computing, 960 Quayside Drive, Suite 406, New Westminster, B. C. CANADA V3M 6G2

Bulletin Board Services

DesignBoard BBS

The DesignBoard BBS is available especially for our customers to access technical support, download updates and technical notes, exchange files and communicate with Capilano personnel.

Access number: (604) 522-4361

Access parameters: 1200/2400/9600 Baud, 8 bits, no parity, 1 stop bit

On your first access you can use basic functions and request validation, or you can phone us in advance and we will create an ID for you.

AppleLink Capilano BBS

Users of the AppleLink network can access the Capilano bulletin board under the Third Party icon for the same services available on our DesignBoard service.

Installer Credits

This program is being distributed using the Apple Installer version 3.3 licensed from Apple Computer, Inc. and *StuffIt* installer technology licensed from Aladdin System. *StuffIt* is a file compression, encryption and file archival utility by Aladdin Systems, 217 E. 86th St., #153, NY, NY, USA.

If you are updating from an older version of DesignWorks

New Features In 3.1

The DesignWorks Simulator version 3.1 contains many important enhancements over the 2.5.x version, including:

• The Simulator now operates on fully hierarchical designs. You can open any subcircuit at any time and probe or modify its internals.

• The Primitive devices have been generalized so that most primitive types can be created in any desired width. E.g. It is now possible to create 17-bit counter or a 128-input AND gate as a single primitive device.

• Pin delays, logic inversion and initial values can be specified individually per pin. This allows path delays through devices to be easily customized.

• Powerful new TestPanel tool allows test vectors to be entered and executed in an interactive, spreadsheet-style format. In addition to straight-forward test vectors, the TestPanel can be used to create a text record of a simulation, to emulate devices or bus interfaces and to monitor simulation results generated by a circuit.

• New "trigger" capability allows automatic detection of specific signal values and time-dependent events such as glitches. Actions including turning the timing display on or off, stopping the simulation and drawing a reference line on the timing diagram can be taken in response to a trigger event.

• The Timing display has many new enhancements, including a timing control panel, bus grouping with hexadecimal value display, text file load and save and many enhanced editing features.

• New simulation model loading and unloading features. The design no longer has to be created and stored with all simulation internal circuits included.

• Input value mapping allows you to specify globally how devices will respond to high impedance and unknown values.

• New display primitive devices include an ASCII display, a 7-segments display and a push-button switch.

• PROM, PLA and RAM devices have been extended. RAM devices can now be loaded from a hex file. PROM and PLA devices are now created from text file input. NOTE: No manual method for PROM or PLA construction is provided in this version.

Updating From 1.x or 2.x

Files created using DesignWorks versions older than 3.0 can be simulated using DesignWorks Simulator version 3.1, but the following points should be noted.

File Format Conversion

DesignWorks 3.x uses a completely new file format and a conversion procedure must be performed on any files created using version 2.5.6 or older. Instructions for this procedure are in the DesignWorks/Converter manual supplied with DesignWorks/Schematic.

Version 3.0 Conversion Errors

The Converter tool supplied with the original DesignWorks 3.0 release did not convert some simulation information in older files correctly. If you wish to simulate these files you must either:

- Reconvert the files from the original 2.x files using the Converter supplied with Schematic version 3.1.1.
- Make a number of corrections to the converted file. These include:
  - Correcting the pin type settings on all "macro" devices in the converted schematic.
  - Verifying the correctness of PLA data. PLA primitive devices marked as Active Low may not be set correctly.
  - Re-initialize RAM data. RAM data was not converted by the original Converter.

Call our Technical Support Department for assistance.

Simulation Setup in Converted Files

Two types of simulation information are not converted when importing older files into version 3.1:

- Timing window setup information: All information about the size and location of the timing window, which signals were displayed and any timing reference lines that were displayed is lost in the conversion process. A new timing window can be created by selecting the Timing item in the Tools menu and then adding traces to it using the Add to Timing command in the Simulation menu.
- Any signal change events that were pending when the file was saved: In particular CLOCK devices will not operate when first opened in version 3.1. The circuit can be restarted by selecting the Clear Simulation item in the Simulation menu. In rare cases, such as a circuit that contains "ring oscillators" or other constructs that require continuous feedback, it may be necessary to manually insert a signal change into the loop using the Signal Probe tool.

Updating from 3.0

DesignWorks 3.0 and 3.1 are fully file compatible and there should be no conversion issues in moving files from version 3.0 to 3.1. Note, however, that the standard symbol libraries supplied with 3.0 and 3.1 contain no simulation information and models will have to be attached to them before they can be simulated.

Refer to Chapter IV - Creating a Schematic for Simulation in the DesignWorks Simulator manual for more information.

This page is intentionally left blank.

DesignWorks™ Simulator

Version 3.0

for the Apple Macintosh™

Reference Manual

May 17, 1993

406 - 960 Quayside Drive
New Westminster, B.C., Canada, V3M 6G2
(604) 522-6200

©1993 Capilano Computing Systems Ltd.

Table of Contents

Chapter I - Introduction 1

Organization of this manual 1
Notes Regarding Copyright 2

Chapter II - Getting Started 3

Installation 3
Installing the Simulation Tools 3
Installing the Device Models 4
Memory Usage 4

Chapter III - Simulation 5

General Information on Simulation 5
Type of Simulation 5
Simulation Memory Usage 6
Time Units 6
Signal Simulation Characteristics 6
Signal States 6
Description of States 7
High Impedance 7
Don't Know 8
Conflict 9
Stuck-at Levels 9
Setting Stuck Levels 10
Clearing Stuck Levels 10
Resolution of Multiple Device Outputs 10
Resistive vs. Forcing Drive 11
Signal Probe Tool 11
Busses 11
Bus Pins 12
Device Simulation Characteristics 12
Device and Pin Delay 12
Primitive Device Delay 12
Sub-Circuit Device Delay 12
Pin Delays 12
Setting the Delay 13
Effect of zero delay 13
Where Delays are Stored 14
Device Storage State 14
Input Signal Values 15
Input Value Mapping 15
Setup and Hold Times 15
Device Pin Types 16
Device Pin Inversion 17

Triggers 18
Simulation Clearing and Initialization 18
The Clear Simulation Operation 18
The Clear Unknowns Operation 19
Setting Initial Values 19
Signal Initial Values 20
Pin Initial Values 20

Chapter IV - Creating a Schematic for Simulation 21

Schematic Simulation Issues 21
Hierarchy Mode 21
Working With Hierarchical Blocks 21
Editing an Open Internal Circuit 21
The Port Interface 22
Parent Device Pin Type 22
Port Connector Pin Type 23
Pin Delays and Inversion 23
Power and Ground Connectors 24
Special Signal Names 0 and 1 24
Simulation Models 25
Primitive Devices on the Schematic 25
Simulation Pseudo-Devices 26
Using External Sub-Circuit Models 26
Using the SimLoad Tool 27
Failure Indications 28
Model Search Folders 28
Selecting Specific Logic Families 29
Batch Loading External Circuit Models 29
Purging External Circuit Models 30
Batch Updating or Reloading Models 30
Model File Naming Conventions 30
Checking Individual External References 31
Customizing Delays for Logic Families 31
Creating External Sub-Circuit Models 31

Chapter V - The Timing Tool 33

The Parts of the Timing Window 33
Displaying Signals in the Timing Diagram 34
Adding a Signal Trace 34
Removing a Signal Trace 34
Repositioning Traces 34
Timing Display Groups 35
Creating a Group Trace 35
Order Within a Group 35
Entering a Group Name 35
The Timing Tool Palette 36

Displaying and Hiding the Tool Palette 36
Tool Palette Time Display 36
Tool Palette Controls 36
Timing Diagram Editing 37
Selecting Data for Copy/Paste Operations 37
Separate Label and Interval Selection 38
Drag Selection 38
Selecting All Traces or All Time 39
Deselecting 40
Summary of Timing Edit Commands 40
The Drawing Tools 41

Chapter VI - The TestPanel Tool 43

General Principles 43
Pop-Up Menus 43
TestPanel Execution Controls 43
Status Display 44
The Data Spreadsheet 44
Data Entry 44
Modes of Operation 45
TestPanel and Hierarchical Designs 45
TestPanel-to-Circuit Association 45
Multiple TestPanels 46
Interaction With Other Simulation Tools 46
Using TestPanel 46
Opening a TestPanel Window 46
Closing TestPanel 47
Editing in the Spreadsheet 47
Editing Text in a Cell 47
Undo 47
Selecting Multiple Cells 47
Moving Between Cells 47
Clipboard Operations 48
Resizing rows and columns 48
TestPanel Controls 48
Step 48
Walk 48
Reset 48
The Test Program 49
Row and Column Usage 49
$Commands 49
Definition Commands 50
Signal List Format 50
The $INPUTS Command 51
The $EXPECTED Command 53
The $OUTPUTS Command 54

The $TIME Command 54
The $WAIT Command 54
The $DELAY Command 55
Control Commands 55
$REPEAT Command 56
$DELAY Command 56
$STOP Command 56
TestPanel Menus 57
File Menu Commands 57
Open Test File 57
Save Test File 57
Set Trace File 57
Save to Design Attr 57
Load From Design Attr 58
Edit Menu Commands 58
Undo 58
Selecting Multiple Cells for Clipboard Operations 58
Exchanging Clipboard Data With The Timing Tool 58
Data Format for Clipboard Operations 59
Cut 59
Copy 59
Paste 59
Clear 60
Select All 60
Fill Down 60
Fill Right 60
Insert Rows 60
Delete Rows 61
Insert Cols 61
Delete Cols 61
Options Menu Commands 61
Link to Circuit 61
Test Options 62
Proceed to Next Step 62
Update Output Values 63
Data Orientation 63
Set Stuck Values 63
Beep at End of Test 63
Clock Width 63
Default Delay 64
Default Header 64
Column Format Options 64
Other Default Header Options 65
Clear Sheet 66
Monitor Mode 66
Copy to Timing 66

Resize Columns 67
Update Display 67
Place and Test Device 67
Save Test to Device 68
TestPanel Examples 68
Testing a Circuit 69
Entering the Test Program 69
Executing the Test Program 70
Dealing With Delays and Setup Times 70
TestPanel as a Pattern Generator 71
Creating Fixed Step Sizes 71
The Test Program 72

Chapter VII - Primitive Devices 73

Gates and Buffers 75
Gate Definition 75
Gate Pin Order 76
Pin Inversions 76
Transmission Gate 77
Three-State Buffer 77
Making Non-Inverting Buffers 78
Resistor 78
Logic Devices 79
Multiplexer 79
Multiplexer Pin Variations 80
Decoder 81
Adder/Incrementer 82
Adder Pin Variations 82
Subtracter/Decrementer 83
D Flip-Flop 83
Flip-Flop Setup and Hold Times 83
Flip-Flop Initialization 84
D Flip-Flop Optional Pins 84
D Latch 84
JK Flip-Flop 85
Register 85
Counter 87
Cascading Multiple Counters 87
Counter Pin Variations 88
Shift Register 89
Clock 89
Creating Synchronized or Offset Clocks 89
Setting Clock Values 90
One Shot 91
Setting One Shot Values 91
I/O Simulation Pseudo-Devices 91

Binary Switch....91
SPST Switch....91
SPDT Switch....92
Logic Probe....92
Hex Keyboard....92
Hex Display....93
SetupHold....93
Unknown Detector....94
RAM Devices....95
Description of the RAM Primitive....95
Don't Know Input Handling....96
RAM Pin Options....96
Creating Custom RAM Devices....96
RAM Size Limits....97
Editing RAM Devices....98
Chapter VIII - Programmable Devices....99
The PROM and PLA Primitive Types....99
PROM Device Format....99
PROM Size Limits....100
PLA Device Format....100
PLA Size Limits....100
Complex Programmable Logic Devices....100
Creating PROMs, PLAs and PLDs for Simulation....101
Creating PROM Devices....101
Generating the Data File....101
The PromPLA Tool....102
Installation....102
Starting PromPLA....102
Editing PROM Devices....103
Reloading PROM Devices....103
Viewing PROM or PLA Contents....104
Creating Programmable Logic Devices....104
The FromABEL Tool....104
Installation....104
Generating the Data File....104
Starting FromABEL....104
The FromABEL Parameters Dialog....105
The FromABEL Synthesis Process....107
Limited or Unsupported PLDs....107
FromABEL Errors....108
Chapter IX - Menu Reference....109
File Menu Commands....109
Open Timing Text....109
Save Timing Text As....110

Print Timing 110
Page Setup 110
Edit Menu Commands 110
Undo 110
Cut 110
Copy 111
Paste 111
Clear 112
Duplicate 112
Point 112
Draw Sig 112
Select All 113
Insert Time 113
Delete Time 113
Simulation Menu Commands 113
Core Simulation Commands 113
Speed 113
Stop 113
Run 113
Other Simulation Speeds 114
Single Step 114
Simulation Params 114
General Delay Box 115
Clock Params Box 117
SetupHold Params Box 118
One Shot Params Box 119
Loading Delays from Attributes 119
Set From Attribute 120
Save to Attribute 120
Triggers 121
Trigger Conditions 121
Trigger Enabling 121
Time Condition Controls 122
Signal Condition Controls 122
Delay Condition Controls 123
Trigger Actions 123
Stick Signals 124
Simulation Scope 125
Clear Simulation 126
Clear Unknowns 126
Signal Probe 126
Probing a Signal 127
Probing a Pin 127
Injecting a Value Using the Probe Tool 128
Timing Window Commands 128
Add to Timing 128

Add as Group 128
Auto Add 129
Find in Timing 129
Timing Options 130
Timing Data Retention 130
Signal Name Display 131
Show/Hide Timing Palette 131
Timing Diagram Pop-Up Menu 131
Get Info 131
Go To Schematic 132
Remove 132
Group 132
Ungroup 132
Collect 132
To Top 132
To Bottom 132

Appendix A - Timing Text Data Format 133

General Description of Format 133
Format Specifics 133
Header Format 134
Single Signal Items 134
Grouped Items 134
Data Line Format 134
Timing Text Example 136

Appendix B - Simulation Attribute Fields 137

Appendix C - Device Pin Types 140

What Pin Types are Used For 140
Pin Types Table 140
Device Pin Type and Simulator Efficiency 141
Bidirectional Pins 142
Output Pins 142
Input Pins 142

Appendix D - Primitive Device Pin Summary 143

Pin Inversion 143
Pin Function Table 143

Credits 147

Index 149

Chapter I - Introduction

The DesignWorks Digital Simulation package consists of a number of tools that work interactively within DesignWorks:

Sim	The digital simulation engine. It is required for all simulation functions.
Timing	The timing diagram display and editing tool. It does not have to be displayed if timing waveform display is not required.
TestPanel	The test vector entry and execution tool. It does not need to be installed if its services are not required.
SimLoad	This tool associates internal simulation models with parts as they are used.
PromPLA	This tool performs data entry and simulation functions for PROM and PLA devices.
RAM	This tool performs data entry and simulation functions for RAM devices.
FromABEL	This tool creates simulation models for devices described using MacABEL.

Organization of this manual

The DesignWorks manual is divided into the following chapters:

Chapter II - Getting Started	Covers installation and initial startup.
Chapter III - Simulation	Details the signal states, time units and other aspects of the simulation. The timing diagram and waveform editing are also discussed.
Chapter IV - Creating a Schematic for Simulation	Discusses signal naming, power and ground, circuit model creation and other issues.
Chapter V - The Timing Diagram	Discusses displaying and editing timing waveforms.
Chapter VI - The TestPanel Tool	Covers the creation and execution of test programs.

Chapter VII - Primitive Devices	Provides information on the simulation characteristics of the primitive device types and the creation and simulation of RAM and connector devices.
Chapter VIII - Programmable Devices	Discusses the creation and simulation of PROM and PLA devices.
Chapter IX - Menu Reference	Provides specific information on the operation of each menu command, except TestPanel menus, which are covered in Chapter VI.
Appendix A - Timing Text Data Format	Describes the text format used for clipboard operations in the Timing tool.
Appendix B - Simulation Attribute Fields	Provides a table of attribute fields used by the various simulation tools.
Appendix C - Device Pin Types	Lists the available pin types and their effect on simulation.
Appendix D - Primitive Device Pin Summary	Provides a summary of pin orders and options for primitive devices.

Notes Regarding Copyright

The DesignWorks Simulator software and manual are copyrighted products. The software license you have purchased entitles you to use the software on a single machine, with copies being made only for backup purposes, unless you have purchase a specific license extension. Any unauthorized copying of the program or documentation is subject to prosecution.

Note regarding trademarks

A number of product trademarks are referred to in this manual. Full credit for these is given in the last section.

Chapter II - Getting Started

This section gives you information on installing and starting up the DesignWorks Simulator.

> NOTE: Since the DesignWorks package and the Macintosh system are constantly being upgraded, there may be recent changes and additional information supplied in Release Notes with the package. It is important that you review these notes prior to installation as they may contain information that supersedes that given here.

We will be assuming that you are already familiar with general Macintosh operation, such as copying files, creating folders, etc. If you are not, then work through the introductory sections of the Macintosh user's guide.

IMPORTANT: Make a backup copy

Before proceeding, make a backup copy and a working copy of your DesignWorks disks, then put the originals away for safekeeping.

Installation

Installing the Simulation Tools

The following table summarizes the code modules making up the Simulator.

Tool	Status	Minimum Memory Usage
Sim	Required	100K
Timing	Optional	115K
TestPanel	Optional	80K
FromABEL	Optional	78K
RAM	Optional	45K
PromPLA	Optional	40K
SimLoad	Optional	33K

In order to be loaded when DesignWorks starts up, these tools must be in the same folder with the DesignWorks program itself, or in the folder specified in the TOOLSFOLDER statement in the DesignWorks setup file, normally "Tools".

Installing the Device Models

Simulation models for the standard digital library components are supplied in the form of "external sub-circuit models", described elsewhere in this manual. These should normally be located in a folder called Models inside the DesignWorks folder.

See the Installation Notes supplied with the package for specific installation instructions.

Memory Usage

Each tool occupies a certain amount of memory space regardless of whether it is being used or not. If you are operating on a system with a minimal amount of memory, it may be desirable to keep unused tools in another folder where it will not be found and loaded by DesignWorks . When it is actually needed, it can be moved temporarily into the Tools folder.

While operating, the Simulator tools can consume large amounts of memory depending upon the size of the design and the amount of data being displayed.

Chapter III - Simulation

This chapter provides more detailed information on the simulation capabilities of DesignWorks.

General Information on Simulation

DesignWorks has the ability to perform a realistic simulation of any digital circuit. Obviously, though, any simulation of any system must be limited in detail and must make certain assumptions. In particular, when simulating digital circuits, it must be understood that real circuits are never completely "digital" in nature, and in fact have many "analog" properties which affect how they operate.

DesignWorks is primarily intended to assist with the logical design of a circuit, and does not take into account factors such as line loading, power supply noise, rise and fall times, output drive, etc. As more of these factors are taken into account, the simulation becomes slower and less interactive, which defeats the purpose for which DesignWorks was created. However, for users with programming capability, the MEDA interface allows programmed simulation models of arbitrary complexity to be added to DesignWorks. These external modules can take into account any desired device characteristics.

Type of Simulation

DesignWorks performs a discrete simulation of the signal changes in a logic circuit, meaning that signal levels and time change only in steps, rather than continuously. The program does not attempt to analyze your circuit, but simply tracks signal level changes through the devices. Thus, circuits with feedback loops or other delay-dependent features will be simulated correctly as long as they don't rely on particular analog characteristics of devices.

The simulation is "event driven", an event being a change in level of a signal. Each time an event occurs, a list is made of all the devices whose inputs are affected by that event. Any other events occurring at the same time are similarly evaluated, and affected devices added to the list. A type-specific routine is then called for each device on the change list in order to determine what output changes are going to occur. These changes are added to the event list, their time of occurrence depending upon the device delay. No computation is performed for times when no event occurs, so that device delay settings and clock values have no effect on how fast the simulation is performed.

DesignWorks performs strictly a digital simulation. It does not take into account factors such as fan-out (i.e. the number of inputs connected to a given output), line length (capacitance), asymmetrical output drive, etc., except in as much as these affect delay time.

Simulation Memory Usage

When a circuit is opened or created by DesignWorks, the circuit data is retained completely in the memory of your machine. Since the total memory available is fixed (until you buy your next memory expansion!), this places some limits on circuit size and simulation.

Each time a signal changes state, an "event" record is created in memory. If the signal is not being displayed in the timing diagram, then this record is deallocated again after the signal change has occurred. If the signal is being displayed, then the record is retained in memory until that change has scrolled off the left-hand side of the timing diagram. As a result, the memory used by event records will increase when the number of displayed signals is increased the resolution of the timing display is decreased or the "retain time" setting is increased.

Time Units

DesignWorks uses 32-bit signed integer arithmetic to calculate all time values used in the simulation. It is usually convenient to think of these values as being in nanoseconds, but the actual interpretation is left up to the user.

The simulation will stop if any time value approaches the 32-bit integer limit.

Signal Simulation Characteristics

Signal States

DesignWorks uses thirteen different device output states in order to track conditions within your circuit. These states can be broken into three groups, as follows.

Forcing States (denoted by suffix .F):

LOW.F

HIGH.F

DONT01.F

DONT0Z.F

DONT1Z.F

CONF.F

Resistive States (denoted by suffix .R):

LOW.R

HIGH.R

DONT01.R

DONT0Z.R

DONT1Z.R

CONF.R

High Impedance:

HIGHZ

Note that the Forcing/Resistive distinction is used only to resolve conflicts between multiple outputs connected to the same signal. The final value stored or displayed for a given signal line can only be one of five possibilities:

LOW

HIGH

DONT

CONF

HIGHZ

Description of States

The High and Low states are the normal ones expected in a binary circuit, but are not sufficient to realistically simulate circuit operation, so the High Impedance, Don't Know and Conflict states are added. There will always be some cases where the simulation will not correctly mimic what would appear in a real circuit, and some of these cases are discussed below. In particular, if a circuit takes advantage of some analog property of a specific device, such as inputs that float high, known state at power-up, input hysteresis, etc., it will in general not simulate correctly.

High Impedance

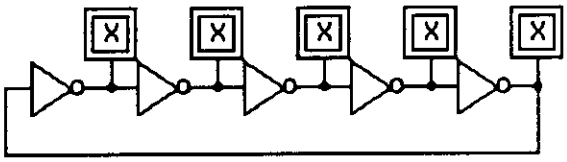
This state ("Z" on a logic probe) is used for cases when no device output is driving a given signal line. This may occur for an unconnected input, or for a disabled "three-state" or "open-collector" type device. If a device input is

in the High Impedance state, it is treated as unknown for the purposes of simulation, even though in a real circuit the device may assume a high or low state, depending on the circuit technology used.

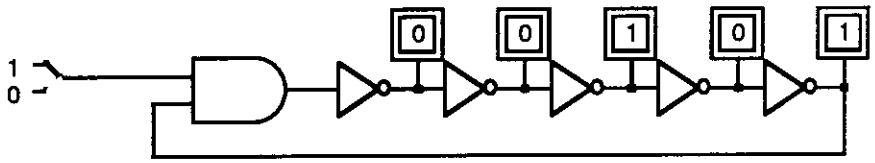
Don't Know

The "Don't Know" state ("X" on a logic probe) results when the simulator cannot determine the output of a device. This may occur, for example, when an input is unconnected or the output from a previous device is unknown. The Don't Know signal will be propagated through the circuit, showing the potential effects of that condition.

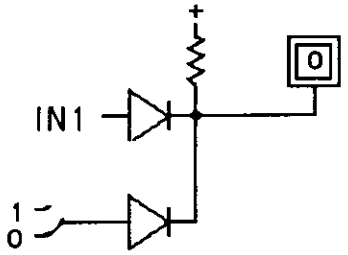
The Don't Know state is used in DesignWorks in cases where the actual result in a real circuit would depend in the circuit technology used, on random chance, or on analog properties of the device not predictable using a strictly digital simulation. For example, if the following ring oscillator circuit is created in DesignWorks, all signals will be permanently unknown, since each depends on the previous one, which is also unknown. In actual hardware, this circuit may oscillate, or may settle into an intermediate logic level, which would not be defined in a digital circuit.



For the purposes of simulation, all circuits must have some provision for initialization to a known state. In most cases, circuits can be initialized using the Clear Unknowns command or by setting the initial value attribute, both described elsewhere in the manual. Alternatively, circuitry can be added to allow a reset to be done, as in the following modification to the ring oscillator:



A problem arises in simulating circuits with multiple open collector devices, such as a bus line, illustrated below:

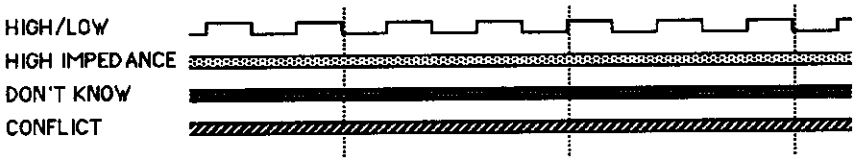


In this circuit the upper device has an unconnected input at IN1 and therefore outputs a "Don't Know" value. The lower device has a low input and therefore outputs a low value. In order to correctly resolve this situation the simulator needs to distinguish between a "Don't Know" output from a normal "totem-pole" type output and a "Don't Know" from an open collector, open-drain, or other single-drive output. In this case the upper device will produce a DONT0Z output which resolves correctly to a LOW on the output regardless of the state of IN1, using the rules described above.

Conflict

The "Conflict" state ("C" on a logic probe) results when two device outputs are connected and are of different or unknown state taking into account the rules described above.

The following timing diagram shows how the various signal states are displayed.



Stuck-at Levels

The DesignWorks simulator implements "stuck-at" levels to assist in setting initial simulation states, testing for faults, etc. When a signal is in a "stuck" state, it will not change state, regardless of changes in devices driving the line.

When the stuck status is set, the signal will retain the value it had at the time until some user action forces a change. When the stuck status is

removed, the signal will return to the value determined by the devices driving the line.

Setting Stuck Levels

A signal can be placed in a Stuck-High or Stuck-Low state by any of the following means:

- applying the name "0" or "1" to the signal.
- typing "H" or "L" while viewing the signal value with the signal probe tool.
- using the Stick High or Stick Low buttons in the Stick Signals command.
- setting a "stuck" option in the TestPanel or other external simulation tool.

Each of these methods is described in more detail in the relevant sections of this manual.

Clearing Stuck Levels

The stuck status can only be cleared by one of the following user actions:

- typing the spacebar while viewing the signal using the signal probe tool.
- clearing the "stuck" state using the TestPanel or other external simulation tool.
- clearing the "stuck" switch in the Stick Signals command

Resolution of Multiple Device Outputs

The DONT0Z and DONT1Z values are used primarily to handle cases of open collector or open emitter devices with unknown inputs (see more information below). Most other types of devices produce the DONT01 output when a value cannot be calculated.

In cases where two or more device outputs are connected together and each one drives the line with a different value, the following rules are used to resolve the actual value on the line.

- the forcing/resistive distinction is only used to resolve outputs from multiple devices. The final value used for display and simulation purposes is one of the forcing values or HIGHZ.
- a forcing drive always overrides a resistive drive or HIGHZ (i.e. the signal takes on the value of the forcing drive, ignoring all resistive drives and HIGHZs).
- a resistive drive always overrides HIGHZ.

- DONT0Z.F and LOW.F produce LOW.
- DONT1Z.F and HIGH.F produce HIGH.
- any other combination of conflicting forcing drives produces CONF.
- DONT0Z.R and LOW.R produce LOW.
- DONT1Z.R and HIGH.R produce HIGH.
- any other combination of conflicting resistive drives produces CONF.

Resistive vs. Forcing Drive

All primitive devices in DesignWorks output a forcing drive level except for the Resistor primitive device. The function of the Resistor device is to convert a forcing drive on one side into a resistive drive on the other. This can be used to modify the output of any existing device type by placing a resistor in series with it. Note that DesignWorks does not model analog properties of devices, so the resistor does not have a resistance value in the analog sense. In particular, there is no interaction between resistor and capacitor symbols to produce delay in lines. The delay effect can be simulated by setting a delay value for the resistor.

Signal Probe Tool

The Signal Probe tool allows you to interactively examine and change values on individual signals and pins in the circuit diagram. When the probe tip is clicked and held on a signal line or pin, the cursor will show the current value on the signal or pin and track changes that occur as the simulation progresses.

See Chapter IX - Menu Reference for more information on the Signal Probe tool.

Busses

Busses (i.e. groups of signals represented by a single line on the schematic) have no particular significance to the simulator. The value of a bus is completely determined by the values of the individual signals it contains. The simulator performs no operations on the bus itself.

NOTE: You can display a bus in the timing diagram using the Add To Timing command, but this is really equivalent to displaying all the internal signals individually and then grouping them.

Bus Pins

Bus pins, like busses, have no particular significance to the simulator. The value of a bus is completely determined by the values of the individual pins it contains. The simulator performs no operations on the bus pin itself. Bus pins are not supported on primitive device types.

Device Simulation Characteristics

Device and Pin Delay

Primitive Device Delay

Primitive devices (i.e. those with a program-defined simulation model) have a single delay value which can be set to any integer value from 0 to 32767. This delay is applied when any input change causes any output change. In addition, a pin delay in the range 0 to 32767 can be set on any input or output pin. Pin delays can be used to set arbitrary path delays through the device. See more information on pin delays below.

The initial delay value is set to 1 when the device is created, but this can be changed later using the Simulation Params command in the Simulation menu. This delay applies whenever any input change causes an output change. There is no provision in the built-in simulation models for different delay values on low-to-high and high-to-low transitions. The Clock and I/O devices have no delay characteristic. See the notes below on delay in hierarchical designs.

Sub-Circuit Device Delay

Sub-circuit devices inherit their delay characteristics from their internal circuit and have no "device delay" characteristic of their own. Simulation Params cannot be directly used on a sub-circuit device, although pin delays can be set separately on each instance of a sub-circuit device to customize path delays.

Pin Delays

Any input or output pin on any device (including port connectors and sub-circuit devices) can have a pin delay associated with it. Pin delays normally default to 0 time units, but can be in the range 0 to 32767.

A pin delay acts like a "buffer" device with the given delay inserted in line with the pin. On an input pin, the device simulation model will not see a change in signal value until after the pin delay has elapsed. On an output pin, the pin delay is added to the overall device delay for any changes scheduled on that pin.

Setting the Delay

Set the delay for a device by first selecting the device (i.e. by clicking the mouse button with the arrow cursor positioned in the device symbol), and then choosing Simulation Params from the Simulation menu.

A dialog box will appear allowing you to increase or decrease the delay value by clicking one of two buttons. The minimum delay value is 0 and the maximum is 32767. When the delay setting for a sub-circuit device is changed, the delays for all internal devices are changed by the same amount.

Effect of zero delay

A delay value of zero is permitted in a DesignWorks device, but this setting should be used only with an understanding of how the simulation is implemented as it can result in unexpected side-effects.

Note that on a given pass through the simulation routine, all the events on the list which occur at the current time are scanned and then the new outputs for all affected devices are calculated. If any of these devices has a zero delay setting then this will result in more changes being placed on the event list at the current time. However, all these changes emerging from zero-delay devices will not be evaluated until the next pass through the simulator. This is done to allow for user interaction with the simulation.

If you step interactively through a circuit with zero-delay elements, you will see all these value changes updated on the screen, even though "simulation time" does not advance. If a signal changes value then reverts to its original state within the same time step, this will be displayed as a zero-width spike on the timing diagram.

If a zero-delay feedback loop exists in a circuit, the signal changes will be simulated and any probes on the diagram will be updated at each pass through the simulator. However the events at the head of the list will always have the same time value associated with them and the simulated time will never advance. This will result in the timing window updating stopping until some delay is inserted in the loop.

Where Delays are Stored

Delays are stored as decimal integers in text form in attribute fields associated with each device or pin. For devices, the delay field is called "Delay.Dev", for pins it is "Delay.Pin" An empty or invalid string will be interpreted as the default value, usually 1 for devices and 0 for pins.

Some special-purpose devices, such as the Clock, One Shot and SetupHold primitive devices take two delay characteristics. In this case, two integers separated by a comma should appear in the Delay.Dev field.

More information on this is given in the information section on each of these primitive types in Chapter VII - Primitive Devices.

A number of additional fields are pre-defined in DesignWorks for storage of alternate delay values. These fields are not used directly by the simulator, but are only provided as a convenient place to store these values. The Save to Attribute and Set from Attribute functions in the Simulation Params command can be used to copy values to or from alternate delay fields.

The following alternate fields are pre-defined.

Field Name	Function
Delay.Dev.Typ Delay.Pin.Typ	Typical device/pin delay
Delay.Dev.Min Delay.Pin.Min	Minimum device/pin delay
Delay.Dev.Max Delay.Pin.Min	Maximum device/pin delay

Device Storage State

In DesignWorks, primitive storage devices (e.g. flip-flops, counters and registers) do not store their current state internally. The device state is completely determined by the values on the signals attached to the output pins. Thus, the following factors will affect the operation of these devices:

- Conflicting or overriding values on the output signals (e.g. a stuck state) will override the last device state calculated by the model.
- Device and pin delays will influence the calculation of a new device state. E.g. if the period of a clock applied to a counter is less than the total delay through it, an erroneous count sequence will result.

If desired, this behaviour can be modified by placing the primitive devices in a sub-circuit device and setting appropriate pin types and delays on the parent device to "buffer" the outputs.

NOTE: These comments do not apply to RAM or bidirectional switch primitives, both of which store internal state information independent of the values of the attached signals.

See the section "Working With Hierarchical Blocks" in Chapter IV - Creating a Schematic for Simulation for more information.

Input Signal Values

Unless an alternate input value mapping is specified (see below), for all device types except switches, signal values "High-impedance" and "Conflict" are treated as "Don't Know" when applied to a device input. When a device is first created, all input signals take the "High Impedance" state and outputs are set depending on their type, normally to the "Don't Know" state. Thus an unused input pin will appear as an unknown input to a device, which may affect its output level.

As with real circuits, all unused inputs should be connected to a high or low level as appropriate. This can be done by naming the pin signal either "0" or "1", using a power or ground symbol, or by using a pullup resistor to set a high level. See more information on logic states in other parts of this chapter.

Input Value Mapping

You can specify an alternate input value mapping for a design by placing a specially-formatted string in the Sim.InputMap attribute field for the design. This is intended primarily to model specific logic families that have a known response to high-impedance inputs. The mapping applies only to primitive devices and is global to the design that it is specified in. Note that this mapping does not occur at inputs to sub-circuit devices but does affect the primitives that they contain.

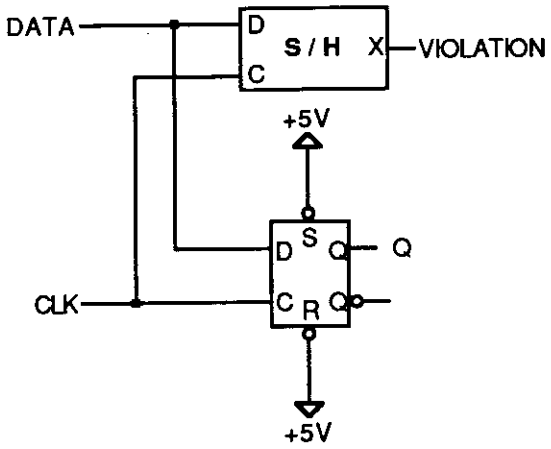
See Appendix B - Simulation Attribute Fields for information on the format of the Sim.InputMap field.

Setup and Hold Times

All standard device types having an edge-triggered clock, such as the D- and JK-flip-flops, register and shift register, have an effective setup time of 1

unit and a hold time of zero units. That is, if the data and clock inputs change simultaneously, the old value of the data input will be used.

Setup and hold times can be checked by attaching a "SetupHold" primitive device to the inputs of the clocked device to be checked, as follows:



The SetupHold device puts out a highZ value until a setup or hold violation occurs when it switches to a don't know state. Thus, the output of the SetupHold device can be paralleled with the flip-flop so that the output line will enter a conflict state when an error occurs.

See more information on the SetupHold device in Chapter VII - Primitive Devices

Device Pin Types

Every device pin has a characteristic known as its *pin type*, e.g. input or output. The pin type is set when the part entry in the library is created and cannot be changed for individual device pins on the schematic. Correct pin type settings are crucial to correct and efficient operation of the simulator.

The pin type is used by the simulator to determine the direction of signal flow and which output values are allowable on a given output pin.

For detailed information on the available pin types and how they affect the simulation see Appendix C - Device Pin Types. For procedures for setting pin types when creating a symbol see Chapter X - Device Symbol Editing in the DesignWorks/Schematic Reference Manual.

Device Pin Inversion

The logic of any pin on any device can be inverted by placing a non-empty value in the Invert.Pin attribute field of the pin. When this is done any value passing into or out from that pin will be inverted. This applies to primitive types as well as sub-circuit devices. The following table summarizes the level mappings that occur.

External Signal Value	Internal Signal Value
LOW.H	HIGH.H
LOW.L	HIGH.L
HIGH.H	LOW.H
HIGH.L	LOW.L
All others	Unchanged

NOTES:

1) The logical inversion of the pin is completely independent of the graphical representation of the pin. I.e. using the "inverted pin" graphic in the DevEditor does not invert the pin logic in the simulator. You must set the Invert.Pin field to have this effect.

2) Although pin inversion can be specified independently for each device on the schematic we do not recommend modifying these settings after a device has been placed on the diagram. This can create the confusing situation of two devices with the same name and symbol but different logical characteristics.

See also:

- *Appendix D - Primitive Device Pin Summary for more information on pin usage and pin inversion.*
- *Chapter VII - Primitive Devices for information on how pin inversion can be used with specific primitive types.*
- *Chapter IV - Creating a Schematic for Simulation for information on pin inversion in sub-circuit blocks.*
- *Chapter X - Device Symbol Editing in the DesignWorks/Schematic Reference Manual for procedures for setting pin attributes when creating a symbol.*

Triggers

The DesignWorks Simulator has a powerful trigger capability analogous to a "word recognizer" on a logic analyzer. Any number of triggers can be set to perform various actions when certain time and signal value conditions are met. These actions can be:

- Drawing reference lines on the timing diagram.
- Stopping the simulator.
- Enabling or disabling the timing display.
- Generating a beep in the computers loudspeaker.
- Enable checking for another trigger.

Any number of triggers can be set up with different activation conditions. For display purposes, triggers are named T1, T2, T3, etc. To assist in setting up complex time-related conditions, trigger T_N can be used to enable T_{N+1}.

See Chapter IX - Menu Reference for specific details on setting up triggers.

Simulation Clearing and Initialization

The DesignWorks Simulator provides a number of mechanisms to assist in setting initial values and restarting a simulation.

The Clear Simulation Operation

The Clear Simulation operation is invoked by either selecting the Clear Simulation command in the Simulation menu or by clicking on the Restart button on the timing tool palette. This operation performs the following steps:

- Other tools (such as Timing and TestPanel) are notified and perform their own processing.
- All signal change events on the queue are disposed of, whether pending or historical.
- Any clocks in the design are re-initialized.
- If any signal or pin initial values are specified, they are set up. See below for information on setting initial values.
- All devices are queued for immediate reevaluation.

The Clear Unknowns Operation

The Clear Unknowns operation is a heuristic procedure which attempts to remove "Don't Know" signal values from a design. This can be used to find an initial state when a design is first simulated or after any edit operations that result in unknown values. This operation is invoked by either selecting the Clear Unknowns command in the Simulation menu or clicking on the Clear X button on the timing tool palette.

The Clear Unknowns operation performs the following steps, stopping as soon as all unknown states are removed from the design:

- Any pending signal change that would result in an unknown state is removed from the queue.
- Any primitive type with storage capability (i.e. flip-flop, register or counter) that has a Don't Know output value is cleared, either to its specified initial value (if any) or to zero.
- The input mapping for the design is temporarily set so that all unknown input values map to zero.
- A single device that currently has an unknown output state is randomly selected and queued for reevaluation. The simulator is cycled repeatedly as long as the number of unknown states in the design decreases. This step is then repeated until the number of unknowns ceases to diminish.
- The input mapping for the design is restored to its original state.

If this operation does not clear the design to an appropriate state, refer to the other techniques discussed below.

NOTE: Design's with "hard" unknowns, such as unconnected inputs or conflicting outputs will not be successfully cleared by this procedure. All device inputs should be specified to a known value if not driven by other devices.

Setting Initial Values

Initial values can be specified for signals and pins which will be applied by the Clear Simulation and Clear Unknowns operations, as described in the preceding sections.

For both object types, the initial value is entered into an attribute field, either Initial.Sig or Initial.Pin. The allowable values consist of a single character chosen from the following table.

Character	Value
0	LOW
1	HIGH
Z	HIGHZ
X	DONT01

All other values will be ignored.

NOTES:

1) It is left completely to the user to decide if the specified initial values make sense. E.g. No checking is done to determine if a given device output value is the reasonable result of the device's current inputs.

2) Devices do not have initial value settings since their values are completely determined by the state of their output pins. See Pin Initial Values below.

Signal Initial Values

The initial value for a signal is stored in the Initial.Sig attribute field using the format described in the previous section. When a Clear Simulation operation is invoked, the initial value specified is placed on the signal without regard for the current output levels of devices driving the signal. The given value will stay on the signal until some device driving the signal changes state or some other user action changes it.

NOTE: If a pin initial value is specified for any output pin driving the signal, the signal value will be overridden.

Pin Initial Values

The initial value for a pin is stored in the Initial.Pin attribute field using the format described earlier. Initial values can only be specified for output or bidirectional pins and will be ignored on input pins.

When a Clear Simulation operation is invoked, the initial value specified is placed on the pin without regard for the current inputs affecting the device. The given value will stay on the pin until the device model schedules a state change or some other user action changes it.

Chapter IV - Creating a Schematic for Simulation

Schematic Simulation Issues

Hierarchy Mode

The DesignWorks Schematic tool supports three hierarchy modes: Flat, Physical and Pure. The following table summarizes these modes in terms of their significance for simulation.

Flat	Despite its name, Flat mode does allow devices to have internal circuits for simulation purposes, although it will discourage access to them. Flat mode also changes the way device names are assigned and reports are generated, but these issues do not affect the simulator.
Physical	This mode allows unrestricted access to internal circuits for simulation purposes.
Pure	Simulation is not supported in Pure mode. This is because only a single definition copy of device type data is kept in memory, regardless of the number of times the device is used. There is nowhere to store the simulation states associated with each instance.

Working With Hierarchical Blocks

The simulator does not impose any new rules on working with hierarchical blocks, but there are some effects of editing a design with active simulation that should be noted.

See also Chapter VII - Hierarchical Design in the DesignWorks/Schematic Reference Manual.

Editing an Open Internal Circuit

A number of issues arise if you have used the same device type or hierarchical block multiple times in a design and you open one copy for editing (i.e. using the Push Into command or by double-clicking on the device). You should note the following points:

• The Schematic tool creates a separate, temporary type definition for the open device when it is opened. Any simulation values that you view or change, or any circuit changes made will apply only to that one device instance while it remains open.

• When you close an open internal circuit, the action taken depends on edits that have taken place. If you have made no edits or changed only instance data (i.e. simulation values or attributes marked as "keep with instance"), then the other blocks of the same type will not be affected by the close. If you have changed any definition data (i.e. any graphical or structural change to the circuit or any definition attribute change) then you will be prompted for the action to take. If you click the Update button, all instance data (i.e. signal values, etc.) from other blocks of the same type will be lost. It will be completely replaced by the values from the edited block.

The Port Interface

The connection between a pin on a parent device symbol and the corresponding signal in the internal circuit is quite complex from a simulation standpoint. In order for this connection to act like a "hard wire" between the two levels, the following conditions must be met:

• the *pin type* on the parent device symbol must be "bidirectional".
• the *pin type* of the corresponding port connector in the internal circuit must be "bidirectional".
• the *pin delays* on both the pin on the parent device and the pin on the port connector must be zero.
• no *pin inversion* must be specified either on the parent device pin or the port connector pin.

Any other combination of settings will result in some degree of isolation or "buffering" between the two levels. I.e. The observed signal value on the signal in the internal circuit may be different from that on the parent pin.

NOTE: When a symbol is created in the DevEditor, all pins default to type "input", i.e. they will not drive any attached signal. If you are creating a hierarchical block symbol for simulation purposes, the pin types must be set to appropriate values.

The effects on these various settings are summarized in the following sections.

Parent Device Pin Type

Any signal value driven out of a parent pin by an internal circuit may be translated according to the pin type on the parent device. These effects are summarized in the following table.

Pin Type	Effect
Input	This will prevent that pin from ever driving the attached signal, regardless of drives in the internal circuit.
Output/Three-state	This will pass the sum of the internal drives up to the parent pin without any translation. Signal value changes on the signal attached to the parent pin will not be passed to the internal circuit.
Open collector/Open emitter	Any drive level from the internal circuit will be translated according the capability of the pin type. See Appendix C - Device Pin Types for more details.
Bidirectional	All changes on the internal signal are passed to the parent pin and vice versa.
Other types	Other types, such as Tied High and Tied Low are not recommended.

NOTE: Although it may be tempting to set all pins to "bidirectional", this is not recommended since it significantly increases simulation overhead and increases the difficulty of isolating circuit drive problems.

Port Connector Pin Type

The pin type on the port connector is also used to translate the value of any incoming signal changes, in a manner similar to the parent pin type. Normally, the pin type setting on a port connector should complement the setting of the parent pin as follows:

Parent Pin Type	Port Connector Name	Port Connector Pin Type
Input	Port Conn In	Output
Bidirectional	Port Conn Bidir	Bidirectional
All others	Port Conn Out	Input

Other settings on the port connector pin are not recommended.

Pin Delays and Inversion

The normal pin delay and inversion settings can be applied to the port interface. A non-null value in the Invert.Pin attribute will cause any signal values passing in either direction to be inverted. An integer value in the

Delay.Pin attribute will cause the specified delay to be inserted in line with level changes passing in either direction.

NOTES:

1) It is recommended that pin delay and inversion settings be applied only to the pin on the parent device and not to the port connector in the internal circuit. Attribute settings on the port connector are more difficult to verify and edit since the port connector is a "pseudo-device" and some schematic editing operations will be disabled.

2) Changes made in Invert.Pin and Delay.Pin after a device has been placed on the schematic, will affect only that one device instance. Default values can be set in these attribute fields when the symbol is created in DevEditor.

Power and Ground Connectors

Power and Ground connector symbols do not have any inherent simulation signal drive unless their pin type has been set to "Tied High" or "Tied Low", as appropriate. Most of the positive-supply symbols provided with DesignWorks 3.1 have "tied high" settings, while others will be "tied low". The symbols provided with older DesignWorks releases may not have any drive setting, resulting in a high impedance level on these signals. This can be remedied by either:

- Replacing any one or all the ground or power symbols with one with the appropriate setting.
- Forcing a Stuck High or Stuck Low level onto the signal using the signal probe tool, TestPanel tool or other external means. Note that because all like-named ground or power segments are logically connected, this only needs to be done on any one segment.

Special Signal Names 0 and 1

The signal names "0" and "1" are recognized by the simulator as special. If any signal is named "0", it will be given a Stuck Low value. If "1" is found, it will be given a Stuck High value. These values can be cleared or changed using the signal probe, if desired.

See the Signal Probe command in Chapter IX - Menu Reference for more information.

Simulation Models

In order to completely simulate a design, every symbol must have an associated simulation model. In DesignWorks, simulation models can take one of the following forms:

- Primitive Devices - These types have "hard-wired" program code to evaluate input and output changes. They include the gates, flip-flops and other devices described in Chapter VII - Primitive Devices, as well as the user-definable PROM and PLA primitives.
- Sub-circuit Devices - The simulation function of a sub-circuit device is completely determined by its internal circuit (except for the addition of pin delays and inversion). The definition of a device sub-circuit can be stored with the part in a library (referred to as an "internal sub-circuit") or can be stored in a separate design file on disk (an "external sub-circuit"). The sub-circuit itself can contain any combination of primitive devices or other sub-circuits (except itself, of course!) nested to any desired depth.
- MEDA Code Models - It is possible to associate an external block of program code with a device for simulation purposes. Whenever any input change or editing event affects the device, the program code will be executed to generate the new output values. This program code can take any desired inputs into account, including disk files, user input, real hardware, etc. Creation of these models requires the optional MEDA Developer's Kit and is beyond the scope of this manual.

Whenever any device type is to be simulated, all information about the device must be loaded into memory. Unless you explicitly purge internal circuits or code models from the design, they will become a permanent part of the design and will be saved with the file.

Primitive Devices on the Schematic

The primitive devices provided in the "Primitive xxx" libraries can be used at any time as part of a schematic, whether or not the simulator is installed. However, these libraries are not intended to match any real logic families and do not have any part name, pin number or packaging information associated with them.

See Chapter VII - Primitive Devices for more information on creating and using primitive types.

Simulation Pseudo-Devices

The simulation pseudo-devices (i.e. those in the Primitive I/O library) are handled specially by the Schematic tool. In general, you cannot modify the symbol, pin types or other characteristics of these devices. In addition, they are treated differently from normal device symbols in the following ways:

- By default, these devices are flagged "omit from report", meaning that they will not appear in any netlist or bill of materials produced by the Report tool. This setting can be changed using the Schematic tool's Get Info command.
- These symbols will not be assigned a name when placed on a schematic. A name can be manually assigned, if desired.
- The switch and keyboard types respond to a normal mouse click by changing state, rather than being selected. To select one of these device, hold the Shift key pressed while clicking on it.

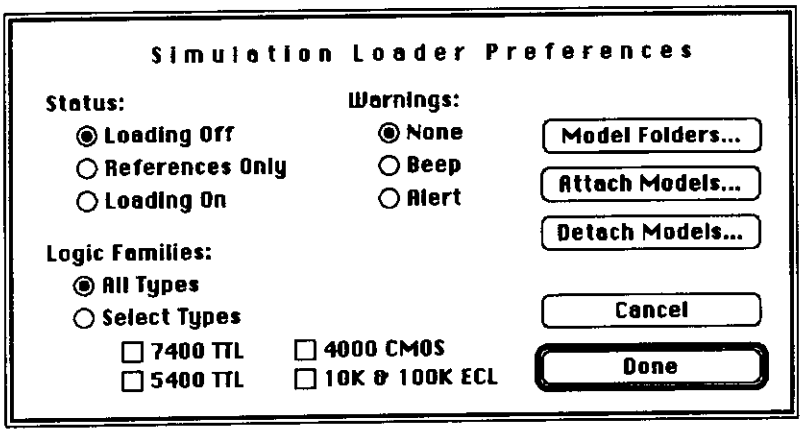
Using External Sub-Circuit Models

Most of the models for industry-standard device types (e.g. 74XX, 4000, etc.) are provided as external sub-circuits. These models are simply design files containing an equivalent circuit for the device. The circuits are designed to match with the symbol libraries provided with the Schematic package.

External sub-circuit models can be loaded either interactively, i.e. as each part is used, or in batch mode after the design is created. Both of these functions are provided by the SimLoad tool.

Using the SimLoad Tool

SimLoad defaults to the inactive state, i.e. no loading occurs. It is activated by selecting its name in the Tools menu. It will then display the following box:



Two levels of loading action are available:

- References Only - This method can be used to check for the existence of the model file without actually loading it. In this case, each time a new part is used, SimLoad attempts to locate a model for it and sets the external circuit reference attributes to the location of the file. The file itself is not loaded, i.e. the device will not have a sub-circuit. The sub-circuit can be loaded later using the Attach External Sub-Circuits command in the Sub-Circuit sub-menu in the Schematic tool's Options menu.
- Loading On - In this mode, SimLoad locates the external model file and immediately loads and attaches it to the part. The external reference attributes are also set, allowing for later updating if the model file is changed.

NOTE: The SimLoad action in both the above cases is the same. In "Loading On" mode, the SimLoad tool commands the Schematic tool to enable its "Auto-Load External Sub-Circuits" mode. This mode can also be controlled directly in the Sub-Circuit sub-menu in the Schematic tool.

Failure Indications

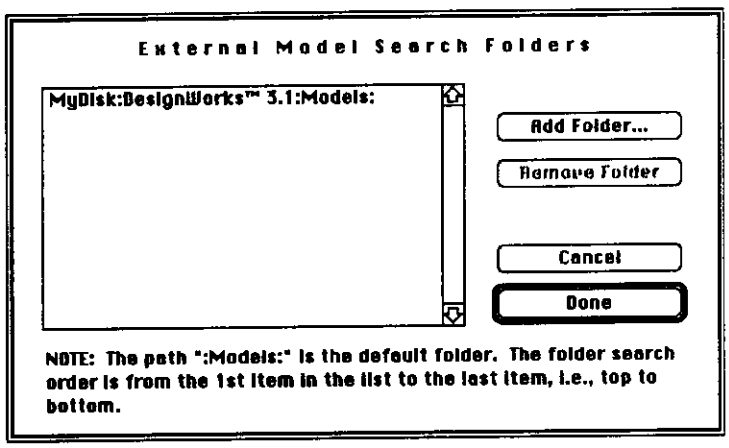
The "Warnings" selections in the SimLoad box allow you to choose the level of warning you receive if a model is not located.

None	No warning is issued.
Beep	A standard beep tone will be issued each time a new device type is used for which a model could not be located.
Alert	A warning box will be displayed whenever a model is not located.

It is possible to determine at any time which devices have an external model associated with them by viewing the contents of the ExtCctName attribute field. This can be done using the Browser tool (select the ExtCctName field in the Secondary field list), or using the Report tool. A special report format file called "External Circuit Form" is provided with the simulator for this purpose.

Model Search Folders

By default, the SimLoad tool assumes that model files are located in a folder called Models in the same folder as the DesignWorks program itself. It is possible to place additional folders on the search list by clicking on the Model Folders button in the SimLoad box. This will display the following box:


External Model Search Folders
MyDisk:DesignWorks™ 3.1:Models:
Add Folder...
Remove Folder
Cancel
Done
NOTE: The path ":Models:" is the default folder. The folder search order is from the 1st item in the list to the last item, i.e., top to bottom.

To add a new folder, click on the Add Folder button. This will display a standard file selection box. In this file box, the "Directory" button adds the directory selected at left to the search list. The "Select Current Directory" button adds the directory containing the current items to the list, i.e. the

directory whose name shows in the file level pop-up menu at the top of the box.

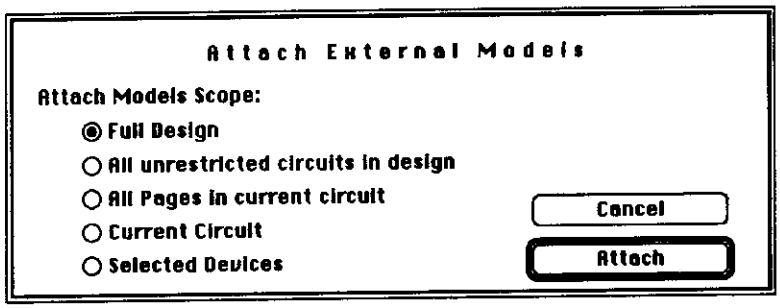
The remove a folder from the search list, simply select it in the list and click on the Remove Folder button.

Selecting Specific Logic Families

The Logic Families selections allow you to restrict the types of devices that will have models attached to them. When the "All Types" button is selected, every device that is used that is a sub-circuit primitive type and does not have an internal sub-circuit will result in a search for a model to attach. When the Select Types button is selected, only the specific standard families indicated will be checked for external models.

Batch Loading External Circuit Models

SimLoad allows the batch loading external models for any group of selected devices or the entire design. The Attach Models button in the SimLoad box will display the following options box:



The Attach Models Scope selections allow you to choose the scope of the search. When the Attach button is clicked, the devices in the given scope are checked, external references are created or updated, where possible, and the external circuits are loaded and attached. After the operation is complete, a summary box will be displayed indicating the number of models found and loaded.

NOTE: This operation can result in a dramatic increase in the amount of memory used by the design. You will be notified if insufficient memory is available to continue.

Purging External Circuit Models

SimLoad's Detach Models operation performs the reverse operation to Attach Models. Devices in the selected scope are scanned and the internal circuit is removed from every device having an external reference. The external reference attributes are left intact, allowing for later reloading of the external models.

Batch Updating or Reloading Models

When an external sub-circuit is loaded, the Schematic tool stores the "Last Modified Date" of the file in the ExtCctDate attribute field of the device. This allows it to check the status of external files relative to a device's internal circuit. Devices that have existing sub-circuits that were loaded from external models, or that have an external reference but no internal circuit, can be updated using the Update External Sub-Circuits command in the Sub-Circuit sub-menu of the Options menu in the Schematic tool. This command will report if all selected items are already up to date.

Model File Naming Conventions

When searching for an external sub-circuit model to attach to a device, the SimLoad tool generates a file name based on the type name, i.e. the name as it appears in the parts palette. The specified search folders are then scanned for the generated file name.

File names are generated differently for each common logic family type, using the following rules:

- File names always end with ".EXT" (note that file names are not case-sensitive).
- Any text in parentheses is ignored, since these are assumed to be package codes.
- If the first two characters of the name are "54" or "74" the a 54XX or 74XX logic family is assumed. Any alphabetic characters are skipped (i.e. the family type) and the file name "74_nnn.EXT" is generated, where "nnn" refers to any remaining digits that are found.
- If the first character of the name is "4" a 4000-series part is assumed. Any alphabetic characters are skipped and the file name "4_nnn.EXT" is generated, where "nnn" refers to any remaining digits that are found.
- If the first character of the name is "1", then a 10K or 100K ECL part is assumed and the name "10_nnn.EXT" or "100_nnn.EXT" is generated, as appropriate.

- For all the above cases, if the name ends with "M", it is assumed to be a "monolith" symbol (i.e. with multiple gates in one symbol) and the M is inserted before the ".EXT", e.g. "74_00M.EXT".
- For all the above cases, if the name ends with a gate unit extension (e.g. "74ALS240.a"), then the unit letter is inserted before the ".EXT", e.g. "74_240a.EXT".
- For all other cases, the type name (with text in parentheses stripped) is used verbatim.

Checking Individual External References

The Attach External Sub-Circuit command in the Sub-Circuit sub-menu of the Options menu in the Schematic tool is useful for checking or editing the external reference attributes of a single device. Select the device in question in the schematic, then select this command. It will show the current status of the external reference and allow you to update it.

Customizing Delays for Logic Families

For cases such as the 74XX families, only one model is provided for all logic families (e.g. 74ALS, 74LS, 74HC, etc.).

> NOTE: These models provide logical functionality only. No attempt is made to match the delay characteristics of the individual parts.

Delays can be manually customized to match a specific part type by either:

- Modifying the sub-circuit model (either in the model file or after it has been used in the design), OR
- Setting appropriate pin delays in the parent device.

Creating External Sub-Circuit Models

External sub-circuit models are simply normal DesignWorks design files with port connectors added to match the pins on the parent device. Following is a brief summary of the considerations affecting sub-circuits:

- In order to be loaded automatically using the SimLoad tool, the file must be named according to the conventions outlined earlier in this chapter.
- A port connector should be placed in the sub-circuit for each pin on the parent symbol, including no-connect pins. Failure to do this will result in the Schematic tool reporting an error each time the sub-circuit is opened and closed.

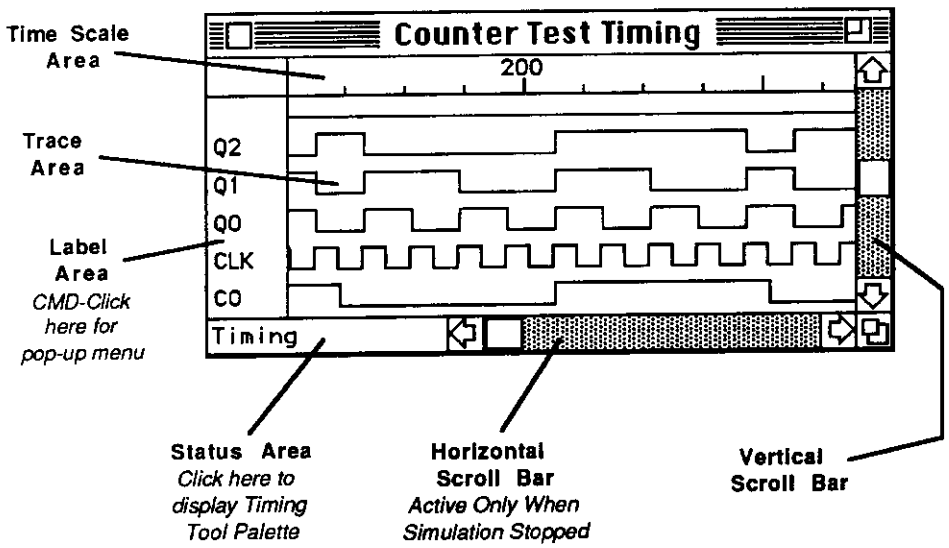
- Each port connector must be named to match the corresponding pin name on the parent device.
- The type of each port connector should match the type of the corresponding pin on the parent device. E.g. An input pin on the parent device should match with a "Port Conn In", or other appropriate type. For more information on pin types, see the section "Working with Hierarchical Blocks" earlier in this chapter.
- The sub-circuit can contain any combination of other devices, including nested sub-circuits to any depth. The only restriction is that the sub-circuit may not contain the parent type or any other type that would create a recursive nesting of blocks.

Other considerations relating to the creation of sub-circuits are discussed in Chapter VII - Hierarchical Design in the DesignWorks/Schematic Reference Manual.

Chapter V - The Timing Tool

The Timing tool allows you to display and edit timing waveforms in graphical form. The timing window is displayed by selecting the Timing command in the Tools menu. Only one timing window can be displayed per design, regardless of the number hierarchy levels in the design.

The Parts of the Timing Window



Time Scale	Located just below the timing window's title bar, the time scale is used to establish the absolute timing of value changes in the trace area. The scale is dependent upon the timing resolution (set using the <> and >< buttons in the timing tool palette). The time scale is also used to set insertion points and selection intervals for use in editing functions.
Trace Area	This area displays simulation results and allows editing of waveforms. Waveforms can only be modified in the future, i.e. times greater than the current simulation time.
Label Area	Displays the list of names of the corresponding to the timing traces at right. Traces can be repositioned by dragging them vertically in this area. In addition, a pop-up trace menu can be displayed by holding the Command (⌘) key pressed while clicking in this area.

Status Area	This box shows progress information for timing commands. Clicking in this box displays the timing tool palette.
Horizontal Scroll Bar	This allows you to display time to the right or left of the present viewing area. The horizontal scroll bar is disabled when the simulation is running.
Vertical Scroll Bar	This will display the signal labels and their corresponding traces above or below the ones presently displayed

Displaying Signals in the Timing Diagram

Adding a Signal Trace

To add one or more signal traces to the timing diagram:

- Select any number of named signals in the schematic.
- Select the Add to Timing command in the Simulation menu.

Removing a Signal Trace

To remove a trace from the timing diagram:

- Hold the Command (⌘) key and click on the name in the label area at the left side of the timing window.
- Select the Remove command in the pop-up menu.

You can remove multiple traces in one operation by holding the Shift key to select multiple labels, then selecting the Remove command;

Repositioning Traces

Any collection of selected labels and their corresponding timing traces can be repositioned within the list by clicking on the desired names (using the Shift key if desired to select more items) and dragging the outlined box vertically to its new location. Releasing the mouse button will cause the list to be revised with the labels and traces in their new positions. Alternaltively, the To Top, To Bottom and Collect commands in the timing pop-up menu can be used.

Timing Display Groups

The Timing tool allows multiple signal lines to be grouped into a single trace with values displayed in hexadecimal.

Creating a Group Trace

A group trace can be created by any of the following methods:

- Select any collection of traces by SHIFT-clicking in the label area, then select the Group command in the timing pop-up menu (i.e. by holding the command (⌘) key while clicking in the label area).
- Select any collection of signals in the schematic diagram, then select the Add As Group command in the Simulation menu.

NOTE: If any of the selected traces is already displayed, it will not be added to the group.

- Select a bus in the schematic diagram, then select either the Add To Timing or Add As Group command in the Simulation menu. Busses are added as a group by default. They can then be ungrouped if desired, using the Ungroup command in the timing pop-up menu.

Order Within a Group

For the purposes of displaying a hexadecimal value for a group, the order of signals within the group is important. When a group is created, the following rules are used to establish the order:

- If the signal name has a numeric part (e.g. "D12" or "WRDAT4X", then the numeric part is used to sort the signals. The lowest numbered signal will be the least significant bit of the group value. Any unnumbered signals will be in the most significant bit positions.
- Otherwise, the signal's existing position is used, i.e. traces that appeared higher in the timing diagram will be more significant.

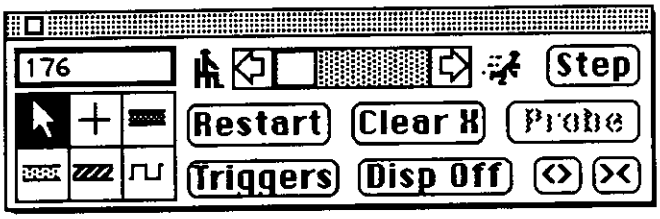
The order of signals within a group can be changed using the Get Info command on a group trace. This is displayed by selecting the Get Info command in the timing pop-up menu or by double-clicking on the label.

Entering a Group Name

When a group is first created, a group name is automatically generated from the names of the enclosed signals. This name can be edited using the Get Info command in the timing pop-up menu.

NOTE: The group name is lost when an Ungroup operation is done.

The Timing Tool Palette



Displaying and Hiding the Tool Palette

The timing tool palette is displayed by default when a new timing window is created. It can be hidden by selecting the Hide Timing Palette command in the Simulation menu or by clicking in its "go-away" box at the top left corner. To redisplay the palette, select the Show Timing Palette command in the Simulation menu or click in the timing status area at the lower left corner of the timing window.

NOTE: The timing tool palette can be displayed even if there is no timing window displayed. This allows you to make use of the simulation controls even if you are not using the timing diagram.

Tool Palette Time Display

The status box at the top left of the tool palette displays one of two different time values, depending upon the status of the simulator:

- If the simulator is stopped and the cursor is positioned in the time scale or trace area of the timing window, it shows the time corresponding to the cursor position.
- Otherwise, it shows the current simulation time as the simulation progresses. It cannot be disabled.

Tool Palette Controls

The control buttons in the tool palette provide quick access to functions alternatively located in the Simulation menu.

...	This speed control allows graphical selection of a simulator speed. Clicking in the gray area of the control switches between Run and Stop. Clicking in the arrows or dragging the position box will select an intermediate speed.

Step	Causes the simulator to execute one time step.
Probe	This item is active only when the schematic is frontmost. This activates the schematic signal probe mode.
Restart	Same as Clear Simulation menu command. Clears all pending events, sets time to zero and recalculates all device states.
Clear X	Same as Clear Unknowns menu command. Clears all storage devices and attempts to clear feedback paths in circuit.
Triggers	Same as Triggers menu command. Displays the trigger control box. See the section on Triggers in Chapter III - Simulation.
Disp Off/Disp On	Enables and disables the timing display. Simulation proceeds substantially faster when the timing display is disabled. It can be re-enabled either manually or by a trigger when desired time or signal conditions are met.
<> (Zoom In)	Increases horizontal display resolution, i.e. decrease number of time units per screen dot.
>< (Zoom Out)	Decreases horizontal display resolution so more elapsed time can be viewed in the display.
Drawing Tools	The drawing tools are used for drawing and editing timing waveforms. These are described in the following section.

Timing Diagram Editing

The drawing tools in the timing tool palette and the commands in the Edit menu can be used to graphically create and edit timing waveforms.

NOTE: Timing traces can only be editing in the future, i.e. at times greater than the current simulation time.

Selecting Data for Copy/Paste Operations

To select timing data for the Cut, Copy, Paste, Clear, Duplicate, Insert Time or Delete Time operations:

- Simulation must be Stopped. To do this, use the speed control in the Timing Tools Palette, select the Stop command in the Simulation Speed sub-menu, or click anywhere in the timing window.

• Cursor must be in 'Point' mode. If not already in point mode click on the Arrow symbol in the Timing Tool Palette, or select the Point command in the Edit menu. The cursor will now be an arrow.

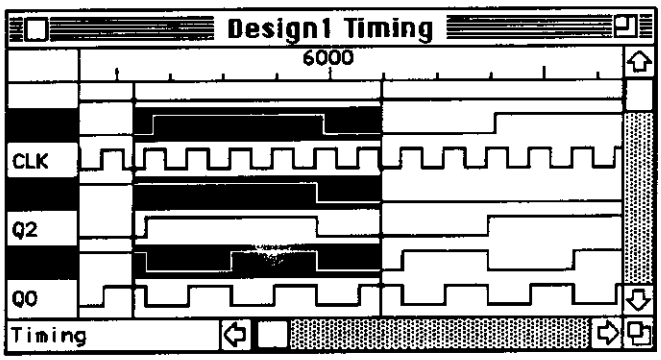
There are two methods of selecting areas for edit operations.

Separate Label and Interval Selection

Using this method, the traces to be affected are selected by SHIFT-clicking in the label area, then the time interval is selected by clicking and dragging in the time scale. This allows you to select non-contiguous traces in the display.

• Click on the desired label in the label area to highlight and select it. To select more than one label, hold the shift key and click on the labels.

• To set the selection interval, click and hold down the mouse button in the time scale at either end of the desired interval. Drag left or right until the desired interval is enclosed. When the mouse button is released the select interval is set and two selection interval lines will appear. If any of the signal labels were selected the timing signal with in the selected interval will be highlighted in time display.



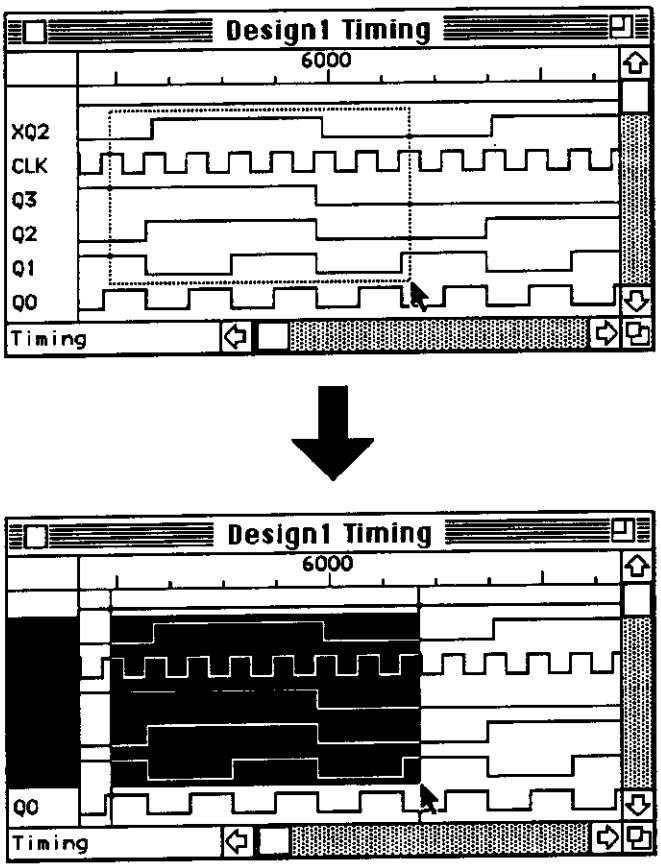
NOTE: Clicking and releasing the mouse button at one spot will create a zero-width interval. This can be used to insert Pasted data without deleting any existing data.

Drag Selection

This method allows you to select a group of labels and a time interval in a rectangular area of the timing diagram.

To do a drag selection, click and hold the mouse button at any corner of the rectangular area you wish to select. Drag diagonally across the desired area.

When the mouse button is released, the enclosed time interval and traces will be selected.



NOTE: The selection operations in the timing window have no effect on selections in the schematic window.

Selecting All Traces or All Time

To select a specific time interval in all traces on the diagram:

- Use the Select All command in the Edit menu to select the entire diagram.
- Drag select an interval in the time scale area without clicking in the trace area.

To select all time for specific traces:

- Use the Select All command in the Edit menu to select the entire diagram.
- Click at the top of the label area, above the highest label displayed (this will deselect all traces), then shift-click to select the desired traces.

Deselecting

Clicking anywhere in the trace area not in a trace will deselect the labels and selection interval.

Clicking in the label area above or below the label list deselects all traces but leaves the current interval selected.

Summary of Timing Edit Commands

The following table summarizes operation of the timing edit commands in the Edit menu.

> *See Chapter IX - Menu Reference for a detailed description of these commands and Appendix A - Timing Text Data Format for a description of the clipboard data format.*

Cut	This command copies to the clipboard any signal change events on selected signals in the selected time interval and clears the selected interval. The data is stored in both picture form and text data form. Events after the selected interval are not moved forward. The Delete Time command can be used to do this.
Copy	The Copy command copies the selected timing data to the clipboard in picture and text format. See the notes under the Cut command, above. Note that Copy can be used on a selection to the left of (older than) the current simulation time since it does not modify the selected data.
Paste	The Paste command pastes the text timing data from the clipboard onto the selected area of the timing diagram. The selected time interval is deleted and then the new data is inserted. I.E. data following the selection interval will be moved forward by the width of the selection interval, then back by the width of the pasted data.
Clear	The Clear command clears any signal change events in the selected area on the timing diagram. Later events are not moved forward.

Duplicate	This command inserts a duplicate of all selected signal data on the timing diagram after the selected interval. I.e. the selected data is copied to a temporary location, then the selection point is moved to the end of the selected interval and the copied data is inserted at this point. All signal changes after the duplicate data are moved back in time by the width of the original selection.
Insert Time	This command inserts a blank time interval in the selected traces. The new interval is inserted in front of the selected interval and is of the same width as the selected interval.
Delete Time	This command deletes the selected time interval from the selected traces and moves all later data ahead by the width of the interval.

NOTES:

1) If you wish to paste a timing picture into a word processing package, it may be necessary to paste it first into a drawing program to select the picture data on the clipboard. A word processing package will normally take the text data from the clipboard by default.

2) You cannot modify timing data that is older than the current simulation time.

The Drawing Tools

The timing tools are used in editing the signal lines displayed in the timing window. New states can only be created and modified in the future (to the right of the point in time where the simulation was halted).

NOTES:

1) The simulator must be stopped to perform any timing editing operations.

2) The drawing tools have no effect on grouped lines. A group can be ungrouped into individual signals for editing, if desired, using the Ungroup command in the timing pop-up menu.

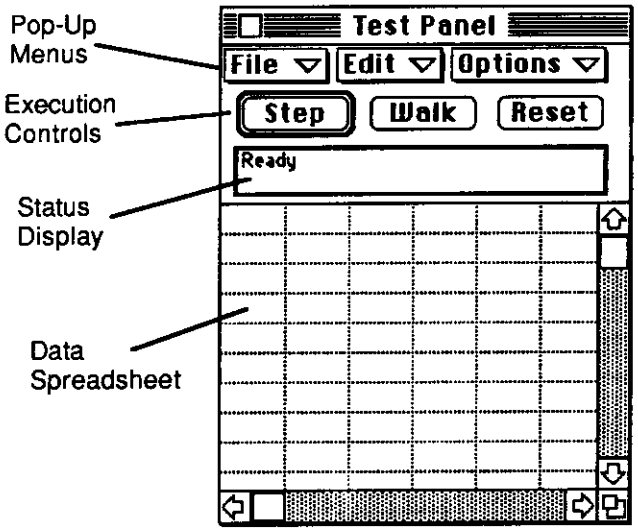
+	**High/Low**	• Creates the a high or low signal state, depending the vertical position the cursor after dragging and releasing the mouse button. • Modifies the time at which a change occurs by clicking on and dragging horizontally the vertical edge of the signal and is set at the time where the mouse button is released. • Modifies the state by clicking on a horizontal edge and dragging up or down. The new state is set when the mouse button is released.
	Don't Know	• Creates the don't know signal state. • Modifies the time at which the state occurs by clicking on and dragging horizontally the vertical edge of the signal and is set at the time where the mouse button is released.
	High Z	• Creates the high impedance signal state. • Modifies the time at which the state occurs by clicking on and dragging horizontally the vertical edge of the signal and is set at the time where the mouse button is released.
	Conflict	• Creates the conflict signal state. • Modifies the time at which the state occurs by clicking on and dragging horizontally the vertical edge of the signal and is set at the time where the mouse button is released.
	Insertion	• Inserts only high/low signal states in between previously created states.

Chapter VI - The TestPanel Tool

The TestPanel tool allows you to interactively step through test vectors (i.e. lists of signal values) and monitor simulation results in a spreadsheet-style format.

General Principles

When the TestPanel is invoked (by selecting its name in the Tools menu), you will see a number controls across the top plus a spreadsheet-like data section across the bottom.



Pop-Up Menus

Each TestPanel window has its own File, Edit and Options menu. TestPanel does not respond to the menus in the main menu bar at the top of the screen.

> *See Chapter IX - Menu Reference for a detailed description of the TestPanel menus.*

TestPanel Execution Controls

The Step, Walk and Reset buttons control basic operation of TestPanel and are described later in this chapter.

Status Display

The status display area shows the current operational status of this TestPanel window.

The Data Spreadsheet

The data spreadsheet section is used to enter your "test program", i.e. a sequence of signal values to apply to the circuit under test, and to view the resulting signal changes.

Time normally progresses downward, i.e. a single row of the spreadsheet represents one simulator time step. Once all these values are set and commands executed, the next line is selected.

> NOTE: The Options command in the Options pop-up menu allows you to select an "across" data orientation. This swaps the two axes of the spreadsheet but otherwise operates identically. To simplify wording, all descriptions of TestPanel operation will assume we are in the default "down" orientation.

Data Entry

Data is entered by clicking in the desired cell and typing directly onto the spreadsheet. Cells can contain signal value data and commands which indicate which signals to deal with and control execution of the test program. Commands always start with a "$" character to distinguish them from data.

In the simplest case, you can enter a sequence of signal values in one column and view the resulting outputs in another column. You can step through the test data interactively using the "Step" or "Walk" buttons, or set up all the events specified in your test program in one block, using the Copy To Timing menu command.

Each of the signal value columns should begin with a "header", i.e. a command listing the signals to be used in that column and whether they are inputs or outputs. If no header is specified in a given column, that column will be ignored and its contents are effectively comments.

Modes of Operation

The TestPanel tool has three basic modes of operation:

- In "interactive mode" (the default), the test vectors and instructions entered in the spreadsheet are evaluated, new signal values are set up in the simulated circuit, and resulting outputs are displayed before proceeding. This mode gives a great deal of interaction and control over the simulation, but at the expense of execution speed. Interactive mode is controlled using the "Step" and "Walk" buttons.
- In "monitor mode", the TestPanel acts as a passive display window for simulation results. In this mode, the circuit is checked after each simulated time step and any desired signal values are displayed in the spreadsheet. Monitor mode is invoked by selecting the Monitor Mode menu command.
- In "copy to timing" mode the data in the data spreadsheet are evaluated and a sequence of signal value changes are set up in the simulator corresponding to all the test vectors. The TestPanel then allows the simulator to proceed to evaluate the scheduled inputs without further interaction. This method of setting up simulation inputs allows the simulator to proceed at full speed, but gives no user interaction during the simulation (of course, the normal DesignWorks interaction with the circuit and the timing diagram are still possible). This process is invoked by selecting the Copy to Timing command.

TestPanel and Hierarchical Designs

TestPanel can be used with hierarchical designs, but each TestPanel window can be associated with only one circuit or sub-circuit at any given time. Multiple TestPanel windows can be open at one time and associated with different sub-circuits. Hierarchical signal names (i.e. prefixed with the names of parent devices) are not supported.

TestPanel-to-Circuit Association

When a TestPanel window is opened, it by default links to the current circuit (i.e. the one displayed in the frontmost document window). It will then search that one circuit for any signal names specified in the test program and perform operation only on that circuit.

It will remain linked to that circuit as long as the circuit is open, even if it is no longer the frontmost. If the circuit is closed, the TestPanel will become inactive until it is attached to a new circuit using the Link to Circuit command in the Options pop-up menu.

NOTE: TestPanel can only be used on circuits that are open for editing.

Multiple TestPanels

Multiple TestPanel windows can be operated concurrently with completely independent operating modes, sets of test vectors and circuit associations. For example, one panel could be used to generate a repeating test pattern while another monitors outputs on a set of signals in a different sub-circuit block.

This is done by simply selecting the TestPanel item in the Tools menu while another TestPanel window is already open. However, we recommend using this technique only once you are quite familiar with general TestPanel operation.

Interaction With Other Simulation Tools

The TestPanel can be used in conjunction with the Timing tool and other simulation display and control mechanisms, however the following points should be noted:

- Results will be unpredictable if two tools are attempting to modify the value of the same signal. This can be caused by the TestPanel modifying a signal that has also been drawn on the timing diagram, multiple TestPanels driving the same signal, or attempting to change the value of a signal that is being driven by a device on the schematic.
- By default, the TestPanel proceeds to the next step in the test program only after all signal states in the design have settled (i.e. there are no signal change events pending). If the design contains any clock devices or oscillations, or if some other tool is creating signal change events, the design may never settle and the next step will never be executed.

Using TestPanel

Opening a TestPanel Window

The TestPanel tool is invoked by selecting the name TestPanel in the Tools menu. The TestPanel window can be left open while working on your schematic, but if any schematic editing is done, the tool will perform an automatic Reset. It is recommended that you close the TestPanel while doing general schematic editing.

Closing TestPanel

You can close the TestPanel window by clicking in the "go-away" box at the top left corner of the window. If any changes have been made in the data spreadsheet and not saved using the File menu commands you will be prompted to save the data.

Editing in the Spreadsheet

IMPORTANT NOTE: The TestPanel has its own pop-up menus for editing commands and does not respond to the commands in the main menu bar at the top of the screen. These will continue to affect the current document window.

Editing Text in a Cell

Text in a cell can be edited at any time simply by clicking in the cell and typing. Standard Macintosh text editing rules apply. Only a single line of text can be typed in a cell.

Undo

Undo is not supported in TestPanel.

Selecting Multiple Cells

Any rectangular block of cells can be selected by clicking in the cell at one corner of the block, then holding the Shift key and clicking at the opposite corner. The selected cells can then be operated on using the commands in the TestPanel's Edit pop-up menu. Drag selection is not supported.

Moving Between Cells

Keyboard operations can be used to move between cells while entering data, as summarized in the following table.

Tab	Moves to the next cell to the right.
Return	Moves to the cell under the last one clicked in.
Arrows	One cell in the given direction.

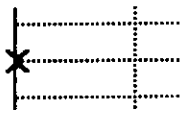
Clipboard Operations

The standard Edit menu operations Cut, Copy, Paste and Clear can be used on any block of selected cells or on a segment of text from a single cell.

See the section TestPanel Menus later in this chapter for a detailed description of these commands.

Resizing rows and columns

Rows and columns of the spreadsheet section can be resized by clicking and dragging on the dividing lines between them anywhere close to the edge of the viewing area. You will see the cursor change shape to a ✕, as in the following:



TestPanel Controls

Step

The Step button causes TestPanel to interactively execute the next line of the data spreadsheet, i.e. the line which contains the currently selected cell. If the line selected is past the last data line, then a beep will be issued and no action will be taken.

Walk

The Walk button causes the TestPanel to step through the data spreadsheet as quickly as possible in interactive mode, starting with the currently selected cell. To start from the beginning of the program, it is necessary to select a cell in the first row of the spreadsheet, or first click the Reset button.

Reset

The Reset button removes all data that was created by the program from the spreadsheet and selects the top-left cell. All user-entered data (i.e. headers, commands, expected values, etc.) are left intact.

The Test Program

The spreadsheet area forming the lower part of the TestPanel window is where all test vector data is entered and results are displayed.

NOTE: The Options menu command allows you to specify an "across" (i.e. time axis is horizontal) or "down" orientation of data in the spreadsheet. The default "down" orientation will be assumed for all discussions that follow. If you are using the "across" orientation, the you will have to swap the terms "row" and "column", etc., in these discussions.

Row and Column Usage

In the spreadsheet area, each horizontal row usually corresponds to one "test vector", i.e. a set of signal stimulus and comparison data that is applied at a particular time step in a simulation. Until the limit of 32767 rows is reached, you can always enter new data into the empty rows displayed at the bottom of the spreadsheet.

The cells in the spreadsheet have no fixed meaning. Their usage at any given time depends upon the last command that was encountered in that column. Normally, we recommend using separate columns for inputs, outputs, and commands, but it is quite possible to mix them in a single column.

$Commands

TestPanel commands always consist of a "$" character followed by a keyword. Commands fall into two groups:

- *Definition Commands* indicate how the following cells in the same column are to be interpreted. These usually give a list of signal names whose values will be specified or observed. Definition commands have no immediate effect on the execution of the test program or on the circuit under test.
- *Control Commands* allow specific loop control, delay or other control actions to be specified.

NOTE: For compactness, command keywords can always be shortened to the minimum number of letters needed to distinguish them from other commands. In most cases, only one letter is needed, e.g. "$I" can be used instead of "$INPUTS". The minimum contraction is shown in the command list below.

Note the following rules concerning command execution:

- Commands have to be "executed" (i.e. stepped through) before they take effect. E.g. if you select the cell under a command and then step downwards from there, the command will have no effect.
- Non-command cells (i.e. anything not starting with a "$") will be ignored completely unless a previous definition command indicates how to interpret them.
- A definition command in a given column always completely overrides any previous definition command in the same column.
- If the circuit schematic is modified while a test program is in progress, all current signal lists are invalidated and no signal data will be interpreted or displayed until the next definition command is encountered.

Definition Commands

NOTE: The $DELAY command can be used either as a control command (if a delay value follows the command keyword) or definition command (if the $DELAY command appears alone in a cell). It appears in both sections below.

Command	Short Form	Description
$TIME	$T	Display current simulation time in the following cells.
$INPUTS signalList	$I	Specify the values of input signals.
$OUTPUTS signalList	$O	View the values of the given signals (can actually be used to view the "real" values of inputs as well).
$EXPECTED signalList	$E	Specify the expected values of the given signals
$WAIT signalList	$W	Pause the test program until a specified value is detected.
$DELAY	$D	Delay by the specified amount before executing this line.

Signal List Format

Many of the definition commands require that a list of signals be specified. The format of this list can be seen in the following examples:

```
CLK
```

specifies simply the single signal "CLK". In this case, each test vector will consist of a single entry, e.g. "0" or "1".

```
A B C D
```

specifies four separate signals A, B, C and D. In this case, each test vector will consist of four values separated by spaces, e.g. "0 1 1 0".

```
[A B C D]
```

uses the grouping operator [] to indicate that the four signals will be specified by a single hexadecimal number with A in the most significant bit position. E.g. the hex test vector "E" (or 1110 binary) would assign a 1 value to A, B, and C, and a 0 value to D.

NOTE: For compatibility with the Timing tool, TestPanel will allow a "group name" to be prefixed to a group, e.g. "CONTROL[A B C D]". This group name is not used internally and will be ignored. For this reason, there must be a blank between any preceding signal name and the opening "[" or the name will be ignored.

```
[D3..0]
```

uses the ".." operator to indicate a sequence of numbered signals. This case also uses the [] to indicate that the four signals D0, D1, D2 and D3 are to be treated as a single integer, in this case with D3 in the most significant position.

```
CLK CLR [D15..0] [FC2..0]
```

specifies four groups, the first being the signal CLK, the second being the signal CLR, the third being the numbered signals D0 through D15, and the last being the set FC0, FC1 and FC2. A typical test vector for this group might be:

```
.C 0 ++ 5
```

In this case, the ".C" specifies that a clock pulse is to be applied to the CLK signal, the value 0 (low) to the CLR, the next sequential integer value to the 16 signals D0, D1, etc. and the hex value "5" (or 101 binary) to the signals FC2, FC1 and FC0. See definitions of these codes below.

The $INPUTS Command

All cells below this command in the same column are used to specify the input values to be set up in the circuit under test. As each line of the test program is read, the cell in this column is interpreted and new values are placed on the signal lines in the circuit. This command must be placed in

the first row of this column to specify which signals are to be set up. The values specified in all subsequent rows are interpreted according to that header. See the information on header format above.

All non-command cells in this column are interpreted as signal value data. A data cell should contain one signal value item for each signal group specified in the most recent header (a signal group being either a single signal, or a group specified in brackets [] in the header. If the cell is empty or if less items are specified in the data cell than in the header, then unspecified items will be left unchanged from the previous time they were specified.

Each signal value item can be one of the items in the following table.

A hexadecimal value	This will be interpreted as an integer with one bit for each signal in the corresponding header group. For individual signals, only the least significant bit is used, so the only values will be 0 or 1. For multiple-signal groups (i.e. specified in []'s in the header), the last signal in the group will take the value of the least significant bit and preceding signals will take the values of successive higher numbered bits. Unused high-order bits are ignored.
The value code ".C" or ".c"	This code indicates a positive clock pulse (i.e. a transition to the 1 level, followed by a transition to the 0 level), with a width as specified in the Test Options command. Note that if the signal is already at a 1 level, no rising transition will occur. See also the notes below.
The value code ".N" or ".n"	This code indicates a negative clock pulse (i.e. a transition to the 0 level, followed by a transition to the 1 level), with a width as specified in the Test Options command. Note that if the signal is already at a 0 level, no falling transition will occur. See also the notes below.
The value code ".Z" or ".z"	This code indicates a high impedance value.
The value code ".X" or ".x"	This code indicates a "Don't Know" value and can be used to test a circuit's response to unknown inputs.
The increment operator "++" or "++n"	This code indicates that the integer value specified previously is to be incremented and the new value applied. If an integer is supplied immediately following the "++" then this number is added to the old input value, otherwise 1 is used.

The decrement operator "--" or "--n"	This code indicates that the integer value specified previously is to be decremented and the new value applied. If an integer is supplied immediately following the "--" then this number is subtracted from the old input value, otherwise 1 is used.

NOTE: Care must be taken regarding setup times when using the .C or .N clock pulse operators. E.g. .C generates a rising transition immediately which may not allow any setup time after values settled from the last step. This situation can be avoided in most cases by using .N to test positive-edge-triggered devices and .C for negative-edge-triggered devices. This ensures that a setup time at least the width of the clock pulse is allowed.

Command examples:

```
$I [D3..0]
$I CLK CLR LD/
$I FC0..2
```

The $EXPECTED Command

This command indicates that the following cells will contain the expected values for signals in the circuit under test. The values in the circuit are compared to those specified. The result of this comparison is shown as a mark character in each data cell this command. If the actual values equal the expected ones, then an "=" is displayed in the cell, otherwise a "≠" is displayed. If no value appears in a data cell, then no comparison is done.

Each signal value item can be one of the items in the following table.

A hexadecimal value	This will be interpreted as an integer with one bit for each signal in the corresponding header group. For individual signals, only the least significant bit is used, so the only values will be 0 or 1. For multiple-signal groups (i.e. specified in []'s in the header), the last signal in the group will take the value of the least significant bit and preceding signals will take the values of successive higher numbered bits. Unused high-order bits are ignored.
The value code ".X" or ".x"	This code indicates a "Don't Know" value and can be used to test a circuit's response to unknown inputs.
The value code ".Z" or ".z"	This code indicates a high impedance value.

The value code ".DC" or ".dc"	This code indicates a "Don't Care", i.e. it will always match. Note that this is different than not specifying a value, in that this will display an "=" result, whereas no comparison is done if no value is given.

The comparison is done either at the same time step that the inputs are set, or after waiting for the simulation to settle (see the Immediately/After Values Settle in the Options command described elsewhere).

Command examples:

```
$E [ADDR0..15]
$E EQ
$E A B C D
```

The $OUTPUTS Command

This command specifies that the following cells should show the actual values read from the circuit and the result of the comparison with the expected values. Any cell in this column not containing a command will be overwritten when the outputs are read.

Command examples:

```
$O [ADDR0..15]
$O EQ
$O A B C D
```

The $TIME Command

This command specifies that the following cells should show the simulation time when execution of this step is started. Time values are displayed in decimal.

Command examples:

```
$TIME
```

The $WAIT Command

This command specifies that the following cells will contain signal value conditions to wait for before proceeding. This command is similar to $EXPECTED, except that instead of display an equal or non-equal status, the test program waits until an equal status occurs. Note the following rules:

- If the wait condition is already met, then the test program proceeds and the wait has no effect.

- The test program enters the wait state after any new input values specified in the row have been set up.
- If any delay is specified in the same row, then the delay applies concurrently and it becomes, in effect, a minimum wait time.

Command examples:

```
$WAIT SYNC
$WAIT [D7..0]
```

The $DELAY Command

This command specifies that the following cells in the same column should be scanned for a decimal integer delay value. The following rules are applied:

- If a value is found, the specified delay is applied after setting up any input value specified in this row. I.e. This specifies the duration of any values applied on this line.
- If multiple $DELAYS appear in one row, then only the longest will apply.
- If no value appears in this column, then the default delay (specified in the Test Options command) will apply.

Command examples:

```
$DELAY
```

NOTE: If any value appears after the $DELAY keyword, then it is taken to be a control command and the following cells in the column will not be interpreted as delays. See the description of this format below.

Control Commands

Command	Short Form	Description
$REPEAT or $REPEAT nnn	$R	Repeat rows up until the next $END either indefinitely or nnn times.
$END	$EN	Marks the end of a REPEAT loop.
$DELAY nnn	$D	Delay for nnn time units after applying the input values specified in this row.
$STOP	$S	Stop execution after this row.

$REPEAT Command

The $REPEAT/$END construct allows you to specify any integer number, or infinite, repetition of a group of test vectors. The format of the $REPEAT command can be seen in the following example:

	$IN CLK	$IN CLR	$IN [D3..0]	$OUT [Q3..0]
	0	0	0	
	0	1	0	
	0	0	0	
$REP 32	0	0	++	
$END	1	0		

In this case, all vectors from the line containing the REPEAT down to and including the line containing the END will be repeated 32 times. If the repetition factor is omitted, it will be taken as infinite in interactive mode, and 1 in timing setup mode.

REPEATs can be nested to any desired depth. If the END is omitted, the last line in the data spreadsheet that has been modified is taken as the end of the loop.

$DELAY Command

When the $DELAY command appears with a value after it, it is taken as a control command and the specified delay is applied after execution of the current row. See the description of the $DELAY definition command above for more details.

Command example:

```
$D 15
```

$STOP Command

This command causes the test program to cease execution after applying any new input values specified in this row.

TestPanel Menus

File Menu Commands

Open Test File

This menu item allows you to load the data spreadsheet from a text file which was saved during a previous session or created externally.

The format of this file is the same as that commonly used for transferring data between spreadsheet applications:

- the text from the top cell in the leftmost column is written first, followed by a tab character, followed by the contents of the next cell to the right, followed by a tab, etc.
- each line is terminated by a carriage return character.

Save Test File

This menu command saves the contents of the data spreadsheet to a text file in the format described above.

Set Trace File

The Set Trace File button allows you to create a text file to which will be written a line of text data matching the current line of the spreadsheet, each time a step is done in interactive or monitor mode. This allows you to create a permanent record of simulation data. This file can be opened and closed at any time during the simulation without disrupting any simulation data.

Save to Design Attr

This command behaves identically to Save Test File As except that the destination for the data is a design attribute field instead of a file. The TestPanel data will then be stored with the design file the next time the design itself is saved.

This command will display a list of the fields defined for the current design allowing you to select one as the destination. All existing data in the

selected field will be lost. Note that attribute fields are limited to 32,000 characters.

Load From Design Attr

This command corresponds to the Open Test File command except that the data is loaded from a design attribute field instead of a file.

Edit Menu Commands

Undo

Undo is not supported in the TestPanel.

Selecting Multiple Cells for Clipboard Operations

The clipboard operations Cut, Copy and Paste can operate on the selected characters in a single cell or on any rectangular group of cells in one operation. To select a group of cells, click at the top-left corner of the group, then hold the Shift key depressed while clicking at the bottom-right corner. All the cells in the group will be highlighted.

NOTE: The Paste command does not require a group to be selected. It always pastes all the data on the clipboard starting at the top-left cell currently selected. The size of the selected area is not significant.

Exchanging Clipboard Data With The Timing Tool

The Timing tool uses a clipboard data format which is a subset of the command structure supported by TestPanel. Therefore, data copied from a timing diagram can always be pasted into a TestPanel window. Conversely, a TestPanel test program can be copied and pasted into the Timing window as long as it is restricted in format. In general, the Timing tool ignores all control commands such as $REPEAT and $WAIT, as well as any expected output specifications.

See Appendix A - Timing Text Data Format for a complete description of the Timing clipboard format.

Data Format for Clipboard Operations

The Cut, Copy and Paste commands operate on text data in the following form:

- text for top-left selected cell, followed by a tab character, followed by text for next cell to the right, followed by a tab, etc. to the end of the selected cells in the row.
- the last cell in a row is followed by a carriage return character instead of a tab.
- if a selected cell is empty, the preceding and following tab characters will indicate its place.

Cut

The action of the Cut command depends upon how data is selected in the spreadsheet. If a group of cells has been selected using the Shift key, then the contents of all selected cells will be copied to the clipboard in the format described above, then the cells will be cleared.

If a cell is being actively edited (i.e. a portion of the text is selected or the I-beam cursor is flashing in it) then only the selected text will be copied to the clipboard and the selected text will be deleted.

Copy

The Copy command operates identically to the Cut command except that the selected items are not cleared.

Paste

The Paste command copies the data on the clipboard to the spreadsheet. Data is assumed to be in the clipboard format described above, i.e. any number of cells can be pasted in one operation.

Note the following rules for the Paste operation:

- The size of the selected area on the spreadsheet has no effect on the Paste operation. The top-left cell in the selected group will be the top-left corner of the area affected by the Paste. However, the number of cells affected by the Paste is determined entirely by the data on the clipboard.
- Tab or carriage return characters cannot be pasted into TestPanel cells since they are used to delimit cell data.

- If a cell is being actively edited at the time of the Paste (i.e. a portion of its text is selected or the blinking I-beam cursor is located in it) then the text on the clipboard up to the first tab will replace the selected text in that cell.
- Data is always copied and pasted as viewed on the screen without regard for the "across" or "down" data orientation TestPanel setting.
- Clipboard operations cannot be undone.

Clear

This command clears all text data from the selected cells. THIS CANNOT BE UNDONE!

> Note: To clear only specific types of cells throughout the spreadsheet area, use the Clear Sheet command in the Options pop-up menu.

Select All

This command selects the entire spreadsheet area.

Fill Down

This command copies the data from the topmost cell in each selected column to all selected cells underneath it. It has no effect on the clipboard.

Fill Right

This command copies the data from the leftmost selected cell in each row to all selected cells to the right of it. It has no effect on the clipboard.

Insert Rows

This command inserts above the topmost selected row a number of rows equal to the number currently selected. All lower rows are moved down.

> NOTE: This affects the entire width of the spreadsheet, regardless of the width of the selected area.

Delete Rows

This command deletes the entire width of each row touched by the current selection or containing the cell currently being edited. All data in the selected rows will be lost and all lower rows will be moved up. THIS CANNOT BE UNDONE!

Insert Cols

This command inserts left of the leftmost selected column a number of columns equal to the number currently selected. The columns to the right are shifted right.

> NOTE: This affects the entire height of the spreadsheet, regardless of the height of the selected area.

Delete Cols

This command deletes the entire height of each column touched by the current selection or containing the cell currently being edited. All data in these columns will be lost and columns to the right will be shifted left. THIS CANNOT BE UNDONE!

Options Menu Commands

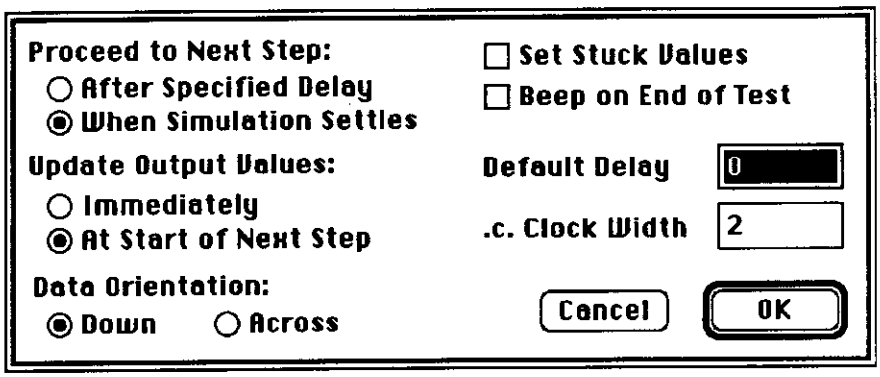
Link to Circuit

This command causes the TestPanel to become associated with the current circuit, i.e. the circuit displayed in the frontmost document window. This is intended primarily for use in hierarchical designs where sub-circuits are being opened and closed.

> *See the notes earlier in this chapter regarding TestPanel and hierarchical designs.*

Test Options

This command will display the following options box:



Proceed to Next Step

These two buttons allow you to determine whether the TestPanel waits for the simulator to complete all processing before proceeding or goes ahead at a fixed pace without regard for other activity.

After Specified Delay	With this setting, the test program will proceed to the next step after the delay specified on that row (if any) or after the Default Delay specified in this box. This setting is most appropriate when using the TestPanel to generate continuous patterns or using it with a design that generates its own clock signals.
After Simulation Settles	With this setting, the TestPanel in effect "goes to sleep" until the DesignWorks simulator has completely settled, i.e. all events have been evaluated and no new ones have been created by the circuit. Note that, depending on the circuit, this may take an arbitrary amount of time. If the simulated circuit has any oscillating elements then this settling will never occur. This setting is preferable for cases where the TestPanel is generating all inputs and clocks, since it allows the test program to be independent of delays in the design.

Update Output Values

These two buttons allow you to determine when the outputs from the design are evaluated.

Immediately	The expected outputs are checked and actual outputs are displayed at the same time step as the inputs were set up, i.e. these will be the values resulting from the previous vector.
At Start of Next Step	Outputs on each line are checked immediately before the next step is executed, i.e. after all simulation values have settled from this step or the specified delay has elapsed. Thus, the output values shown on each row will be the result of the inputs specified in that row.

Data Orientation

These two buttons allow you to select whether time proceeds downwards (i.e. each row represents one time step) or across (each column represents one time step).

If this setting is changed, you will be asked whether you want to swap all data in the spreadsheet area.

Except for the reversing of row and column functions, all TestPanel operations are identical in either mode.

Set Stuck Values

When this checkbox is set, all signal values set up by the TestPanel are "stuck", i.e. they override any outputs from existing devices in the circuit and can only be changed by further TestPanel outputs or other user settings.

Beep at End of Test

When this switch is enabled, a standard beep tone will be issued whenever the end of the test program is reached.

Clock Width

This editable text item determines the width of the clock pulse generated by a ".C" or ".N" signal value in the "Specified Inputs" column. A zero value is not allowed.

Default Delay

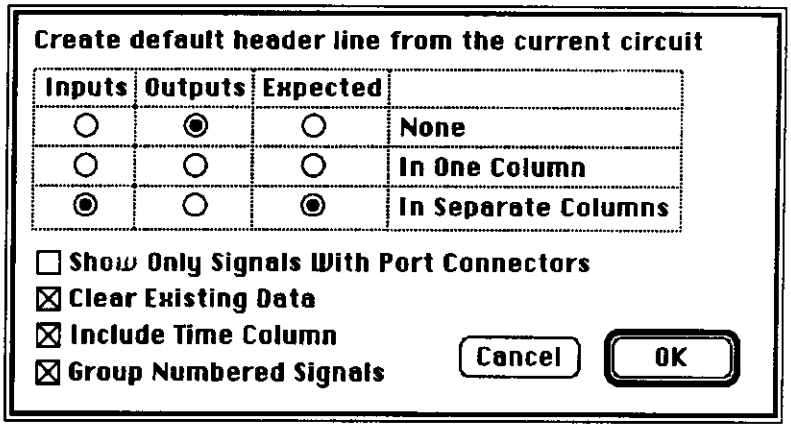
This editable text item operates slightly differently depending on other settings:

- If the "After Specified Delay" step option is selected or a "Copy to Timing" operation is being done, this will be the default delay between steps and can be overridden by a $DELAY command in the test program. A value of zero is not allowed.
- If the "After Simulation Settles" option is selected, this will be the minimum step size. I.e. The time between steps will be the larger of this value or the amount of time required for the circuit to settle. A value of zero is allowed in this mode.

Default Header

This command examines the current circuit (i.e. the frontmost circuit window) and finds all undriven inputs and final outputs in the circuit and creates a default header in the first row of the spreadsheet area. These can then be edited manually if needed.

The following options box will be displayed:



Column Format Options

The nine buttons in the Inputs/Outputs/Expected grid allow you to select how header entries are created for input and output signals.

NOTE: "Outputs" and "Expected" both refer to the output signals, but offer different treatment of them. It is allowable to included the same signals in both places.

These options are summarized in the following table.

	Inputs	Outputs	Expected
None	No input columns ($INPUT) are included	No actual output columns ($OUTPUT) are included	No expected output columns ($EXPECTED) are included
In One Column	All inputs are included in a single column, e.g. $I A B C D	All outputs are displayed in a single column, e.g. $O A B C D	All expected outputs are checked in a single column, e.g. $E A B C D
In Separate Columns	Individual inputs or groups are shown in separate columns	Individual outputs or groups are shown in separate columns	Individual outputs or groups are checked in separate columns

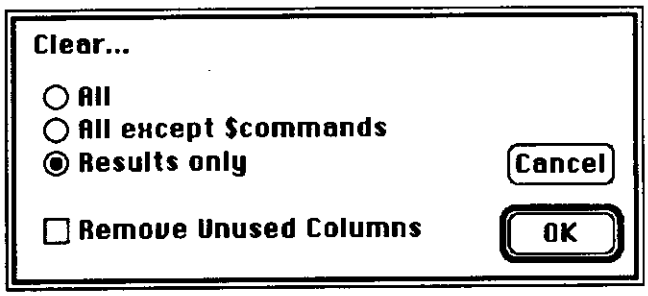
Other Default Header Options

The following table summarizes the remaining options in the Default Header box.

Show Only Signals with Port Connectors	This option is intended for use with sub-circuit blocks that have Port Connector symbols attached to all external pin connections.
Clear Existing Data	If this box is checked, all data currently in the spreadsheet area will be cleared before creating the new header. THIS CANNOT BE UNDONE!
Include Time Column	This option will insert a column at the left with a $TIME header. This will display time values as the test program executes.
Group Numbered Signals	When this option is enabled, sequentially numbered signals (e.g. FC0 FC1 FC2) will be grouped together using the "[]" grouping operator, e.g. "[FC2..0]". They can then be specified with a single hexadecimal value. The lowest numbered signal is assumed to be the least significant bit of the value. This applies to all Inputs, Outputs or Expected columns.

Clear Sheet

This command clears the entire data spreadsheet area, offering the following options:



All	All cells in the spreadsheet are cleared
All Except $Commands	Any cell starting with a "$" character will be left untouched.
Results Only	All cells that were filled in by the program will be cleared and all others left untouched.
Remove Unused Columns	If this box is checked, any columns that are completely empty will be deleted.

Monitor Mode

This command selects monitor mode operation. In this mode, any command or specification that would set up a signal value change in the circuit under test is ignored. Each time any signal specified as an output changes state, a new line is displayed in the spreadsheet area.

Selecting this command again returns to normal mode.

Copy to Timing

This button performs the "timing setup mode" conversion of test vector data to simulation events. If the DesignWorks simulator is not stopped, then simulation will proceed immediately after events are set up.

NOTES:

1) The meanings of the some of the commands in the Command column of the spreadsheet are changed in this mode. For example, an infinite repeat loop is not allowed and will be interpreted as a single iteration.

2) This command does not affect the clipboard.

Resize Columns

This command adjusts the width of all columns so that the widest data in the column just fits.

Update Display

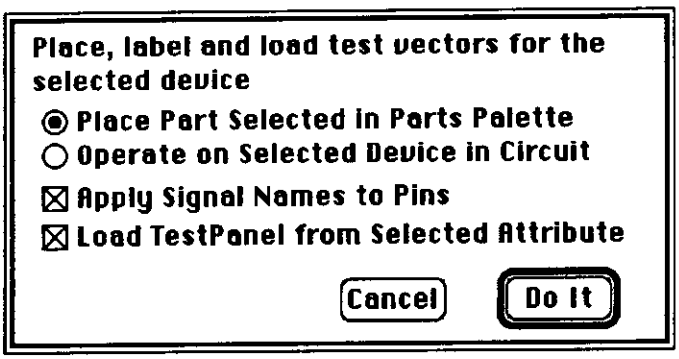
When this box is checked, the spreadsheet is updated with new data after each operation. This provides best user feedback on the progress of the test, but substantially slows down the process. This control only affects the appearance of the display. All other controls still operate interactively. This switch can be turned on and off at any time during operation to check on progress of the test.

Place and Test Device

This command provides the ability to load a device from a library and ready it for testing in a single operation. It is intended to allow a quick verification or analysis of a library part.

NOTE: Most of the parts in the Primitive libraries shipped with the Simulator contain test vectors which can be used to verify their operation or as examples for creating your own device test procedures. Simply click on the part in the Parts Palette once so that it is selected, then select this command.

This command will operate either on a part type selected in the currently displayed library in the Parts palette, or it will operate on a selected device already placed in the current circuit. It displays the following option box:


Place, label and load test vectors for the selected device
Place Part Selected in Parts Palette
Operate on Selected Device in Circuit
Apply Signal Names to Pins
Load TestPanel from Selected Attribute
Cancel
Do It

Place Part Selected in Parts Palette	If this button is selected, then the part currently selected in the Parts Palette will be placed in the center of the current schematic page. The circuit is assumed to be empty, but no checking is done.
Operate on Selected Device in Circuit	If this button is selected, then the operations selected below will be performed on the device that is currently selected in the schematic.
Apply Signal Names to Pins	If this option is checked, a visible signal name will be applied to each pin, corresponding to the name of the pin. This name can then be referred to in the test vectors.
Load TestPanel from TestVectors Attribute	If this option is selected then the spreadsheet area will be cleared and new data loaded from the TestVectors.Dev attribute field in the device.

Save Test to Device

This command causes the entire spreadsheet area to be saved to the TestVectors.Dev attribute field of the device selected in the schematic.

See the comments under the Save to Design Attr command for the format and limitations of this data.

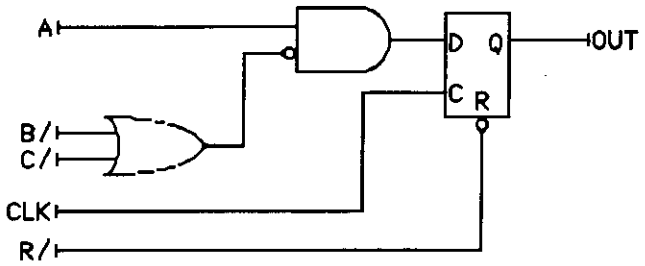
TestPanel Examples

This section provides some simple examples of TestPanel usage. For more complex examples, see the example files provided with the simulator release.

NOTE: Most of the devices in the Primitive device libraries contain stored test vectors and can be used in conjunction with the Place and Test Device command in the Options pop-up menu for example purposes. See the description of this command elsewhere in this chapter.

Testing a Circuit

In this example we will execute a set of test vectors for the following simple circuit made up of primitive devices:



This example and the test program below are provided as files with the Simulator release.

Entering the Test Program

NOTE: This file is included with the Simulator release and can be loaded using the Open Test File command in the File pop-up menu.

With the above circuit open in the frontmost schematic window, follow these steps:

- Select the TestPanel item in the Tools menu. An empty TestPanel window will appear.
- Select the Default Header command in the Options pop-up menu.
- For this example the "Include Time Column" switch was turned off, although you can optionally leave it on and shift all entries right one column.
- Click OK on the Default Header box. This will create the header entry for the first row.

• Enter the remaining cells as shown below. Note that it may be convenient to enter the first few lines manually then use the Fill Down command in the Edit pop-up menu to set the default values in the remaining rows.

$I R/	$I CLK	$I C/	$I B/	$I A	$E OUT
0	0	1	1	0	0
1	0	1	1	0	0
1	.N	1	1	0	0
1	.N	0	1	0	0
1	.N	0	0	0	0
1	.N	0	0	1	1
1	.N	1	0	1	0

We are now ready to execute the test program. You may wish to save the current entry using the Save Test File As command in the File pop-up menu.

Executing the Test Program

To execute the test program:

• Click on the Reset button. This will select the top-left cell in the spreadsheet.
• Click on the Walk button. If the simulator is Stopped, you will be prompted to start it, otherwise the program will be executed.

If all TestPanel and delay settings are the default values, this program will fail in the last step, due to a setup time problem. This will be explained in the next section.

Dealing With Delays and Setup Times

In this example, we are feeding inputs through a combinational logic array to an edge-triggered device. We must be aware that a delay must be allowed between changes at the combinational inputs and the application of the clock edge to the flip-flop.

In this case, the flip-flop is positive-edge-triggered. We have used the ".N" negative clock pulse notation since this in effect inserts a delay equal to the pulse width before the rising edge. This is because is creates a negative edge first (which is not significant to this flip-flop), then waits for the specified pulse width, then inserts the positive edge.

The default TestPanel clock pulse width is 2 units and the default delay in primitive devices is 1 unit. The last test vector in the above program causes

a logic transition to pass through two gates, for a total delay of 2 units. Thus, the input transition at the D input to the flip-flop arrives at the same time as the positive edge of the clock. The D flip-flop model always uses the old value at the input, so the new value does not reach the Q output.

There are two solutions to this problem:

- Change the TestPanel clock pulse width using the Test Options command in the Options pop-up menu. This will affect all clock pulses generated by this TestPanel window.

NOTE: These settings are not saved with the test file and will return to default values the next time the program is restarted.

- Change the test program to set the clock input explicitly and insert a delay at this line, as in the following changes to the final line:

1	.N	0	0	1	1
$DEL 5	0	1	0	1	.DC
1	1	1	0	1	0

In this case, we are replacing the .N operator with two separate test vectors with explicit 0 and 1 values for the clock and a delay of 5 in between.

TestPanel as a Pattern Generator

TestPanel can also be used as a repeating test pattern generator or even as a replacement for a device or bus interface which is not present in the circuit design.

In this case, we will use the $WAIT command to synchronize the values generated by the TestPanel with the design under test. When the SYNC signal enters the 0 state, the D0..3 lines will be set to zero. When SYNC goes to 1, the TestPanel will cycle through a Gray code sequence and then loop back to the beginning.

Creating Fixed Step Sizes

By default, the TestPanel executes each step when the simulation settles after the previous one. This is appropriate when the test program is driving the circuit and evaluating the outputs, but in this case we want a fixed step size, regardless of other simulation activity. To do this:

- Select the Test Options command in the Options pop-up menu.
- Enter the value 10 into the Default Delay box.
- Select the After Specified Delay option.
- Click OK on this box.

The TestPanel will now proceed at fixed intervals whether or not the simulation has settled.

The Test Program

The following test program performs the described function.

	$WAIT SYNC	$I [D3..0]
$REP	0	
	1	0
		1
		3
		2
		6
		7
		5
		4
		C
		D
		F
		E
		A
		B
		9
$END		8

As a variation on this, removing the 0 and 1 entries from the $WAIT column or deleting the column altogether, will cause the test program to loop continuously without synchronization.

Chapter VII - Primitive Devices

This chapter provides information on the "primitive" or built-in device types in DesignWorks. These types are intended primarily for use in creating model circuits for higher-level macro devices. Because their simulation functions are hard-coded, they occupy much less memory space than macro devices and simulate more efficiently.

IMPORTANT NOTES:

1) In primitive devices, the association of logic functions to pins on a device symbol are made by pin order. When creating primitive devices using the DevEditor tool, you must be aware of the pin order requirements for the device type you are using. Refer to the description of each type in this chapter and to Appendix D - Primitive Device Pin Summary.

2) Bus pins are not supported on primitive device types.

The following table lists the available primitives and their functions.

Primitive Type	Description	Related Type	Max. # Inputs
NOT	Inverter		1
AND	N-input AND gate	Any pin inversions	799
NAND	N-input NAND gate	Any pin inversions	799
OR	N-input OR gate	Any pin inversions	799
NOR	N-input NOR gate	Any pin inversions	799
XOR	N-input XOR gate	XOR	799
XNOR	N-input XOR gate	XNOR	799
Transmission Gate	Transmission gate		1
Buffer	Non-inverting N-bit 3-state buffer with optional common inverted enable	Buffer	400
Resistor	Digital resistor		1
Multiplexer	M*N-to-M multiplexer		256
Decoder	1-to-N line decoder		256
Adder	N-bit adder with carry in and out	Incrementer	256
Subtractor	N-bit subtractor with borrow in and out	Decrementer	256

D Flip-Flop	D-type flip-flop	optional S & R	1
JK Flip-Flop	JK flip-flop	T flip-flop, optional S & R	1
Register	N-bit edge-triggered register		256
Counter	N-bit synchronous counter	Up/down	256
Shift Register	N-bit shift register		256
One Shot	Retriggerable one shot		1
Clock	Clock oscillator		1
Binary Switch	Debounced toggle switch		1
SPST Switch	Open/closed single pole switch		1
SPDT Switch	Double throw switch		1
Logic Probe	Signal level display		1
Hex Keyboard	Hexadecimal input device		1
Hex Display	Hexadecimal digit display		1
SetupHold	Setup/Hold time checker		1
Unknown	Unknown value detector		1

The following table lists devices supported primarily for compatibility with older version of DesignWorks. We do not recommend using these in new designs.

Device	Description
Pullup	Pullup resistor, single pin
D Flip-Flop ni	D-type flip-flop (non-inv S & R)
JK Flip-Flop ni	JK flip-flop (non-inv S & R)
Glitch	Glitch detector
SimStop	Simulation halt device

Gates and Buffers

The Primitive Gates library contains the primitive gates with a built-in simulation function. The NOT, AND, NAND, OR, NOR, XOR and XNOR devices behave according to the appropriate truth tables for such devices. Any gate input which is in the "Don't Know", "High impedance" or "Conflict" state is treated as a "Don't Know". A gate with a "Don't Know" input will not necessarily produce a "Don't Know" output. For example if one input of an AND gate is low, the output will be low, regardless of the state of the other input, as in the following truth table:

A	B	OUT
0	0	0
0	1	0
0	X	0
1	0	0
1	1	1
1	X	X
X	0	0
X	1	X
X	X	X

Gate Definition

The gate types, except NOT, can be created with any number of inputs (from 0 to 799) and are defined as shown in the following table.

Function	Output is...	Output is DONT if...
AND	LOW if any input is LOW, HIGH otherwise	Some input is DONT and no input is LOW
NAND	HIGH if any input is LOW, LOW otherwise	Some input is DONT and no input is LOW
OR	HIGH if any input is HIGH, LOW otherwise	Some input is DONT and no input is HIGH
NOR	LOW if any input is HIGH, HIGH otherwise	Some input is DONT and no input is HIGH
XOR	HIGH if an odd number of HIGH inputs and no DONTs	Any input is DONT
XNOR	HIGH if an even number (or zero) of HIGH inputs and no DONTs	Any input is DONT

Gate Pin Order

The NOT type must have exactly one input and one output, in that order. All other logic gate types can have any number of inputs followed by a single output, up to the maximum DesignWorks limit of 800 pins.

NOTE: Pin order is important in all primitive devices! When creating a gate type using the DevEditor tool, the output pin must be the last item on the pin list.

Pin Inversions

The logic of any pin on any device can be inverted by placing a non-empty value in the Invert.Pin attribute field of the pin.

For example, to create the following AND gate with one inverted pin:

the following steps must be taken in the DevEditor tool:

1) Create the desired graphic symbol using the DevEditor's drawing tools.
2) Place the three pins as shown. ORDER IS IMPORTANT! All primitive devices must have a specific pin order. For gates, all inputs come first and the output pin last.
3) In the pin list, double-click on the last pin (the output pin). This will display the Pin Info palette for that pin.
4) Set the pin type to Output, and, if desired, edit the pin name.
5) Click the Enter key on the keyboard to move to the next pin and edit the other pin names, if desired, and check that they are all set to Input.
6) Click the close box in the Pin Info palette.
7) In the pin list, click once to select the input pin that is to be inverted, then select the Pin Attributes command in the DevEditor menu.
8) Select the All Fields option in the attributes box, then select the Invert.Pin field.
9) Enter the value "1" for this field, then click Done. (The actual value doesn't matter, as long as it is non-empty.)
10) Select the Sub-circuit and Part Type command in the DevEditor menu.
11) Click on the Primitive Type button, then select the AND primitive type in the pop-up menu.
12) Close the Part Type box and save the part to a library in the usual manner.

NOTES:

1) The logical inversion of the pin is completely independent of the graphical representation of the pin. I.e. using the "inverted pin" graphic in the DevEditor does not invert the pin logic in the simulator. You must set th Invert.Pin field to have this effect.

2) Inverted gate types NAND and NOR can be created by using the NAND and NOR primitive type setting or by using the AND and OR settings and inverting the output pin (or the input pins, using DeMorgan's Theorem). The two methods will produce identical simulation results. There is a slight memory overhead but no execution speed overhead to using an inverted pin.

Transmission Gate

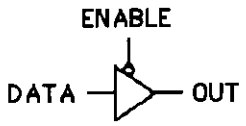
The transmission gate (X-Gate) device behaves as an electrically controlled SPST switch. When the control input is high, any level change occurring on one signal pin will be passed through to the other. Since it has no drive capability of its own it will behave differently than a typical logic device when a high-impedance or low drive level signal is applied to its signal inputs. Most other primitives, such as gates, interpret any applied input as either High, Low or Don't Know. The transmission gate, on the other hand, will pass through exactly the drive level found on its opposite pin. Thus, a high-impedance level on one pin will be transmitted as a high-impedance level on the other pin. Note that the simulation of this device may produce unpredictable results in extreme cases, such as an unbroken ring of transmission gates.

NOTE: No variations in number or order of pins are possible with the XGATE primitive type. It must have exactly one control pin and two bidirectional pins with pin order as described in Appendix D - Primitive Device Pin Summary.

Three-State Buffer

The three-state buffer has N data inputs, N data outputs, and an optional active-low enable input. If the enable input exists and is high, all outputs enter a "High-impedance" state. If the enable input doesn't exist, or is low, each output will follow the corresponding input if it is low or high, or produce a "Don't Know" level otherwise.

A single-input three-state buffer is shown in the following table:



ENABLE	DATA	OUT
0	0	0
0	1	1
1	0	Z
1	1	Z

Making Non-Inverting Buffers

The Buffer primitive type can also be used to make a non-inverting buffer, i.e. with its outputs always enabled, simply by omitting the enable input.

This can be used for the following purposes:

- To represent a non-inverting buffer or level translator in a design.
- To insert a delay in a signal path without affecting the logic of the signal.
- To create various types of open collector, open emitter or inverting buffers, when used in conjunction with different pin type and inversion settings on the outputs. NOTE: It is more efficient to use the NOT primitive type to make a simple inverter.

Resistor

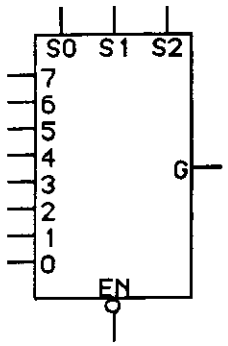
The resistor device simulates the effects of a resistor in a digital circuit. It is more general than the Pullup Resistor device and can be used as a pullup, pulldown or series resistor. Whenever a signal level change occurs on either pin of the resistor, the device converts that level into a resistive drive level (see Chapter III - Simulation for more information on drive levels). A high impedance drive on one end is transmitted as a high-impedance output to the other end. Note that DesignWorks does not simulate analog properties of devices so the resistor device does not have a resistance value in the analog sense and will not interact with capacitor symbols placed on the same line. The effect of resistance on line delay can be simulated by setting the delay of the resistor device.

Logic Devices

Multiplexer

This is a device which selects one of N data inputs and routes it to a corresponding output line. There can from 1 to 256 outputs, plus an optional enable input, as long as the total pin count does not exceed the 800 pin limit.

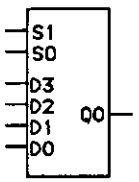
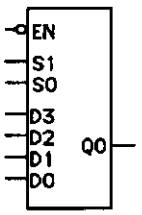
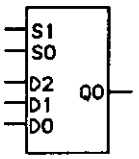
A typical 8-to-1 multiplexer obeys the following function table. X = Don't Care.



EN	S2	S1	S0	7	6	5	4	3	2	1	0	G
0	0	0	0	X	X	X	X	X	X	X	0	0
0	0	0	0	X	X	X	X	X	X	X	1	1
0	0	0	1	X	X	X	X	X	X	0	X	0
0	0	0	1	X	X	X	X	X	X	1	X	1
0	0	1	0	X	X	X	X	X	0	X	X	0
0	0	1	0	X	X	X	X	X	1	X	X	1
0	0	1	1	X	X	X	X	0	X	X	X	0
0	0	1	1	X	X	X	X	1	X	X	X	1
0	1	0	0	X	X	X	0	X	X	X	X	0
0	1	0	0	X	X	X	1	X	X	X	X	1
0	1	0	1	X	X	0	X	X	X	X	X	0
0	1	0	1	X	X	1	X	X	X	X	X	1
0	1	1	0	X	0	X	X	X	X	X	X	0
0	1	1	0	X	1	X	X	X	X	X	X	1
0	1	1	1	0	X	X	X	X	X	X	X	0
0	1	1	1	1	X	X	X	X	X	X	X	1
1	X	X	X	X	X	X	X	X	X	X	X	1

Multiplexer Pin Variations

A number of variations in multiplexer logic are possible with this primitive type, depending on which input and output pins are included. The following table summarizes the possible variations. Samples are shown with M=1 and N=2, but any combination of M and N can be used within the maximum pin limit of 800.

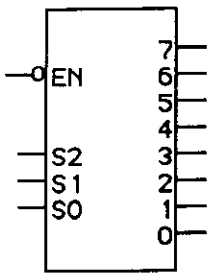
Number of Sections	Number of Inputs/Section	Number of Select Inputs	Number of Enable Inputs	Sample Symbol
M	2^N	N	0	
M	2^N	N	1	
M	2^N-1+1 .. 2^N	N	0*	

* If there are less than 2^N inputs per section there can be no enable input.

Specific pin order requirements for the multiplexer type are given in Appendix D - Primitive Device Pin Summary.

Decoder

The Decoder activates one of N outputs, depending on M select inputs, as follows (X = Don't Care):



EN	S2	S1	S0	7	6	5	4	3	2	1	0
0	0	0	0	1	1	1	1	1	1	1	0
0	0	0	1	1	1	1	1	1	1	0	1
0	0	1	0	1	1	1	1	1	0	1	1
0	0	1	1	1	1	1	1	0	1	1	1
0	1	0	0	1	1	1	0	1	1	1	1
0	1	0	1	1	1	0	1	1	1	1	1
0	1	1	0	1	0	1	1	1	1	1	1
0	1	1	1	0	1	1	1	1	1	1	1
1	X	X	X	1	1	1	1	1	1	1	1

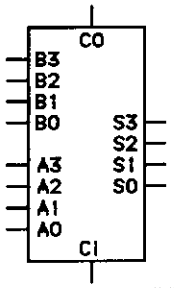
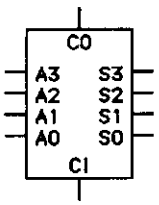
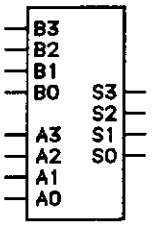
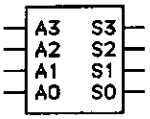
Adder/Incrementer

The N-bit Adder accepts one or two N-bit input arguments and (optionally) a 1-bit carry and outputs their N-bit sum plus an optional 1-bit carry out.

Multiple adders can be connected together by feeding the carry out from each stage to the carry in of the next more significant stage. The carry in to the least significant stage should be set to zero.

Adder Pin Variations

The adder primitive can be used in four variations, as summarized in the following table. Sample symbols are shown with 4 bit inputs, although any number of bits from 1 to 256 is permissible.

	Has B Inputs	No B Inputs
Has Carry In	S = A + B + Cin 	S = A + Cin 
No Carry In	S = A + B 	S = A + 1 

In addition, the Carry Out pin can be independently included or omitted from any of these configurations.

Refer to Appendix D - Primitive Device Pin Summary for precise pin order requirements.

Subtracter/Decrementer

The Subtracter primitive type behaves identically to the Adder type except that a subtract or decrement operation is performed, depending upon pin configuration.

D Flip-Flop

The D-type flip-flop is positive-edge triggered and obeys the following function table:

S	R	D	Clock	Q	Q/
0	0	X	X	1	1
0	1	X	X	1	0
1	0	X	X	0	1
1	1	0	Rises	0	1
1	1	1	Rises	1	0
Rises	Rises	X	X	X	X

In the above table X on the input side means "Don't Care" and on the output side means "Don't Know".

Flip-Flop Setup and Hold Times

None of the DesignWorks primitive types explicitly implement variable setup and hold times. However, all edge-triggered primitives have an effective setup time of 1 unit since they always use the input signal value existing before the current step. I.e. if the data input changes at the same time as the clock, the old data value will be used to determine the new output value. This effective setup time can be modified by specifying input pin delays on either the data or clock pins.

Checking for setup and hold violations can be done in one of two ways:

- Attaching a SetupHold primitive device to the inputs and output of the device under test. The combined device could be made into a sub-circuit if desired to combine it into a single symbol.
- Using the Trigger capability of the simulator to watch for value changes within a set amount of time.

Flip-Flop Initialization

Note that when a flip-flop is first placed in the schematic, it is in an unknown state and must be correctly initialized before it will produce predictable outputs. This can be done in the following ways:

- Adding circuitry to force an explicit reset.
- Using the Clear Unknowns command in the Edit menu to force an initial state before starting the simulation.
- Specifying an initial output value for both the Q and Q/ outputs in their respective Initial.Pin attributes. This will be applied every time a Clear Simulation command is executed.

D Flip-Flop Optional Pins

The D Flip-Flop primitive type has the following optional pins:

- The Q/ (Not-Q) output can always be omitted.
- The Set input alone or both the Set and Clear inputs can be omitted.

Refer to Appendix D - Primitive Device Pin Summary for specific pin order information.

D Latch

The D Latch primitive type is identical to the D Flip-Flop in function and pin specifications except that it is level-triggered instead of edge-triggered. I.e. The Q and Q/ outputs will follow the level of the D input as long as C is high.

JK Flip-Flop

The JK flip-flop is negative-edge triggered and obeys the following function table:

S	R	J	K	Clock	Old Q	New Q	New Q/
0	0	X	X	X	X	1	1
0	1	X	X	X	X	1	0
1	0	X	X	X	X	0	1
1	1	0	0	falls	0	0	1
1	1	0	0	falls	1	1	0
1	1	0	1	falls	X	0	1
1	1	1	0	falls	X	1	0
1	1	1	1	falls	0	1	0
1	1	1	1	falls	1	0	1
rises	rises	X	X	X	X	X	X

In the above table X on the input side means "Don't Care" and on the output side means "Don't Know".

If any inputs are in an unknown state, the simulator will determine the output state where possible, or else set it to "Don't Know".

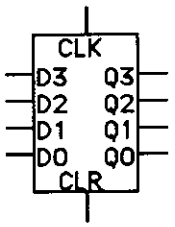
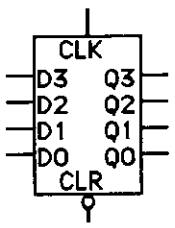
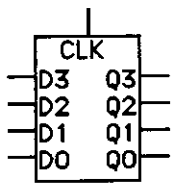
See the notes under D Flip-Flop, above, on setup and hold times and initialization.

Register

This device implements an N-bit, positive-edge-triggered register with common clock and an optional active-high clear inputs.

See the comments on Setup and Hold times and initialization in the D Flip-Flop section, earlier in this chapter.

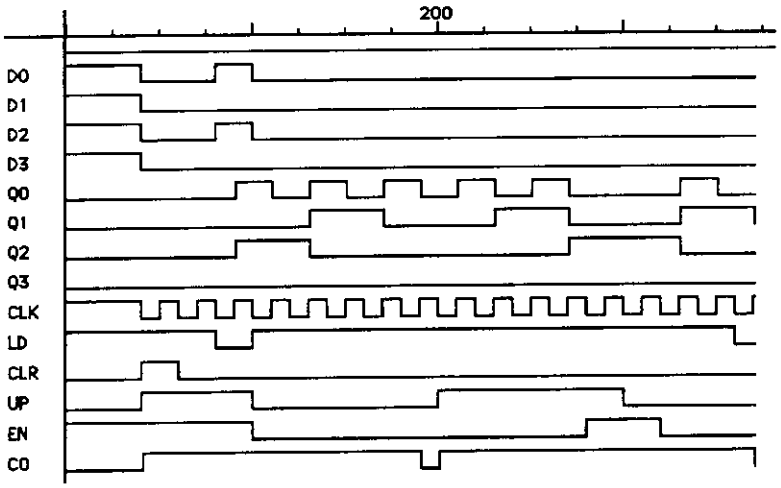
The following table illustrates some pin variations available for the Register primitive type.

4-bit register with active-high clear	
4-bit register with active-low clear (using pin inversion)	
4-bit register without clear	

Counter

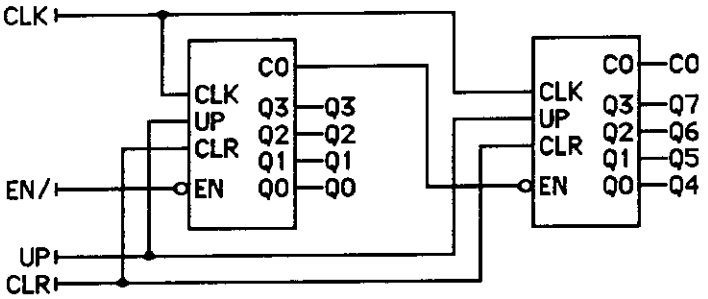
This device implements a N-bit, presettable, synchronous, positive-edge-triggered up/down counter with active-low enable. The load data inputs and most of the control inputs can be omitted for simplified versions.

The following timing diagram shows a typical count cycle. Note that the CO (Carry Out) output goes low when the count reaches 2^N-1 (when counting up) or 0 (when counting down) and rises again on the next count. This can be used to cascade multiple counters together, as shown. The CLR input clears the counter asynchronously (i.e. regardless of the state of the clock). The Count/Load input, when low, causes the data from the N data inputs to be passed to the outputs on the rising edge of the next clock. The Enable input disables counting when high, but has no effect on loading.



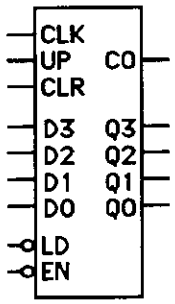
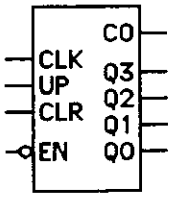
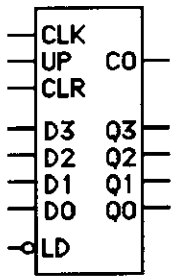
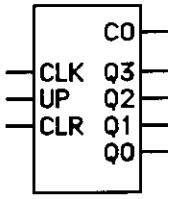
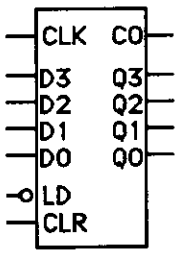
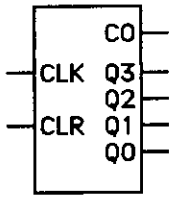
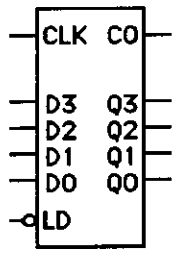
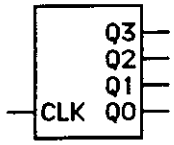
Cascading Multiple Counters

Counter primitives with the optional Enable and Carry Out pins can be cascaded to form larger synchronous counters as follows:



Counter Pin Variations

The following table summarizes the possible pin usage variations for the counter primitive type. The sample are shown with N=4, although the number of bits can be anywhere in the range 1 to 256.

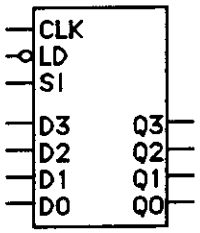
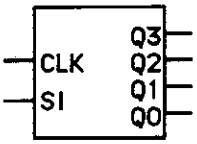
Optional Inputs	Including Load Inputs	Excluding Load Inputs
CLR, UP/DN, ENABLE		
CLR, UP/DN		
CLR		
none		

NOTE: COUT can be independently included or omitted in any of the above variations.

Shift Register

The shift register is an N-bit, positive-edge-triggered device with serial or optional parallel load. When the Shift/Load input is low, data from the N parallel data input lines is transferred to the outputs on the rising edge of the next clock. When Shift/Load is high, the next rising clock edge causes the value at the Shift In input (SI) becomes the new value for output Q0, Q0 shifts to Q1, Q1 to Q2, etc., and the old value at the most significant output is lost.

The following table shows the shift register primitive with and without parallel inputs.

With Parallel Load	Without Parallel Load
	

NOTE: The Shift Register primitive cannot be created without data outputs (i.e. a parallel-in serial-out register) because the flip-flop values are stored on the output pins. Primitive devices have no internal state storage. See more comments on this in Chapter III - Simulation.

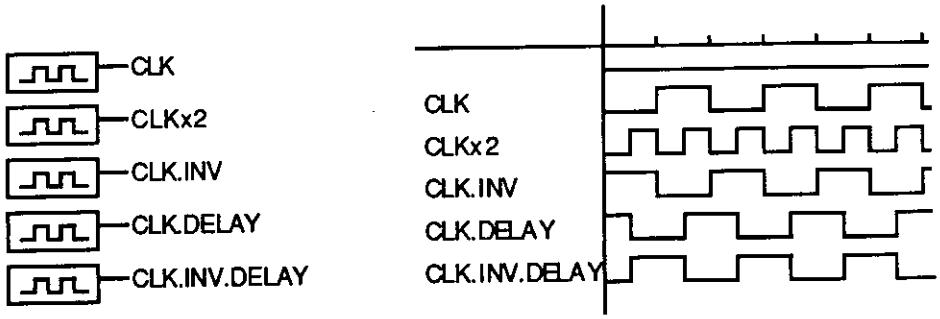
Clock

The clock oscillator is used to generate a repeating signal to activate other devices. When it is first created, the clock output pin will be low, then after a delay time called the "Low Time" it will change to the high state. After a further delay called the "High Time" the signal will revert to low, and the cycle will repeat. The low and high times are initially set to 10, but can be modified using the Simulation Params item in the "Simulation" menu. Any number of Clocks may exist at once with independent delay times.

Creating Synchronized or Offset Clocks

When the Clear Simulation command is selected (or the Restart button on the Timing palette pressed), all clocks in the design are restarted. Clock outputs will be set to the low state and the timer for the low period will be restarted. Clock high and low times, combined with pin inversion and pin

delay settings can be used to precisely determine the relationship between two clock outputs. The following circuit example summarizes these options.



Signal	Low Time	High Time	Invert.Pin	Pin Delay
CLK	10	10		0
CLKx2	5	5		0
CLK.INV	10	10	1	0
CLK.DELAY	10	10		5
CLK.INV.DELAY	10	10	1	5

Setting Clock Values

To set the high and low times for a clock, first select the device in question (by setting the cursor to "Point" mode and clicking the mouse button inside the device symbol), then choose the **Simulation Params** item in the Simulation menu.

You will be presented with a dialog box with buttons for increasing or decreasing the high and low values. The minimum for either value is 1 and the maximum is 32767.

See more information on the Simulation Params command in Chapter IX - Menu Reference.

One Shot

The one shot is used to generate an output pulse of a fixed length when it is triggered by the rising edge of the trigger input. Two parameters can be set for a one shot: the delay from the rising edge of the input to the start of the output pulse, and the duration of the pulse. The delay and duration times are initially set to 1 and 10, respectively, but can be modified using the Simulation Params item in the "Simulation" menu.

The One Shot device is retriggerable, meaning that the output pulse will not end until "duration" time units has passed since the last trigger input. Repeating the trigger input can cause the output pulse to be extended indefinitely.

Setting One Shot Values

To set the delay and duration times for a one shot, first select the device in question (by setting the cursor to "Point" mode and clicking the mouse button inside the device symbol), then choose the Simulation Params command in the Simulation menu.

Refer to Chapter IX - Menu Reference for more information on the Simulation Params command.

I/O Simulation Pseudo-Devices

Binary Switch

The Binary Switch device provides a means for setting a signal to a low or high level. When a switch is first created its output is at a low level. Pointing at the switch with the cursor in "Point" mode and clicking the mouse button causes the switch arm to move and the output to change to the opposite state. Any number of device inputs can be driven by a switch output. A switch has no delay characteristic since it has no inputs.

SPST Switch

The SPST switch device simulates the actions of a simple open/closed switch in a digital circuit. When a switch is first created it is open and both connections present a high impedance logic level. Clicking on the switch with the cursor in "Point" mode causes the switch arm to close and the switch to "conduct". In terms of the digital simulation, this means that whatever logic level is present on each pin is transmitted to the other one.

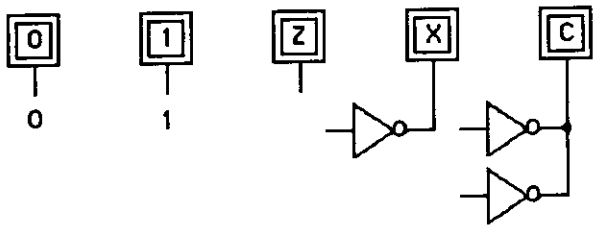
An SPST switch has a default delay of zero but this can be set to any value from 0 to 32767 using the Simulation Params command. The Shift key must be pressed on the keyboard in order to select a switch device.

SPDT Switch

The SPDT switch device operates in essentially the same manner as the SPST switch described above, except that it is always conducts between the single pin on one side and one of the two pins on the other. As with the other two switch types, clicking on it with the Point cursor changes the position of the contact.

Logic Probe

The logic probe is a device for displaying the level present on any signal line. When the probe is first created, its input is unconnected and therefore in the "High-impedance" state, which will be displayed as a "Z". When the input pin is connected to another signal, the displayed character will change to reflect the new signal's current state. Any further changes in the signal state will be shown on the probe. Possible displayed values are 0 (low), 1 (high), X (Don't Know), Z (High impedance) or C (Conflict).



Hex Keyboard

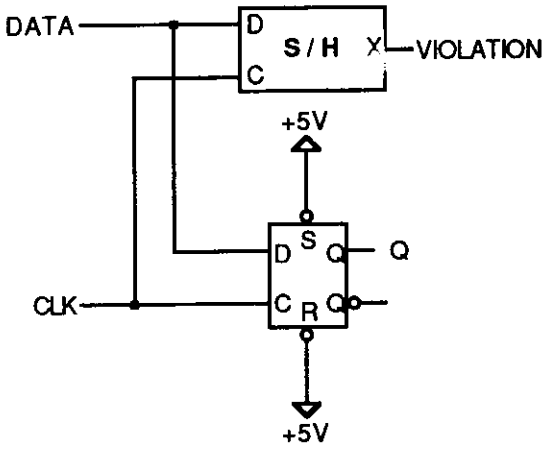
The hex keyboard outputs the binary equivalent of a hexadecimal digit on four binary lines. A "key" is pressed by positioning the tip of the arrow cursor in the desired key number and clicking the mouse button. The binary data on the output lines will change to reflect the new value and remain until the next key is pressed. On a fifth line, it outputs a signal which goes high momentarily and then low again when a key is pressed.

Hex Display

The hex display shows the hexadecimal equivalent of its four binary inputs. If any of the inputs is unknown, high-impedance or conflict, then an X will be displayed.

SetupHold

Setup and hold times can be checked by attaching a "SetupHold" primitive device to the inputs of the clocked device to be checked, as follows:



Each time the "D" input changes, the SetupHold device marks the current time. On the next rising edge of the "C" input, the current time is compared to the last transition on the "D" input. If the difference is less than or equal to the Setup time specified in the Simulation Params box, the output switches to a Don't Know state and remains in this state until a non-violating input transition occurs. Similarly, for each clock rising edge, the time is marked and compared to the time at the next transition on the "D" input. If this difference is less than the Hold time given, the output is set to Don't Know.

The SetupHold device puts out a highZ value until a setup or hold violation occurs when it switches to a don't know state. Thus, the output of the SetupHold device can be paralleled with the flip-flop so that the output line will enter a conflict state when an error occurs.

Unknown Detector

The Unknown Detector device detects any unknown value (i.e. not 0 or 1) on its input and puts out a high value after the delay time specified in the Simulation Params box. When the input returns to a 0 or 1 state, the output becomes zero. This can be used in conjunction with the SimStop device to detect invalid conditions in a circuit and stop the simulator.

RAM Devices

The RAM primitive device supports the direct simulation of static Random Access Memory devices in a variety of configurations. A number of common RAM types are provided in the Primitive RAM library. In addition, the RAM tool allows you to create custom RAM devices with a variety of pin options.

NOTE: To create or simulate RAM devices, the RAM tool must be installed in the Tools folder.

Description of the RAM Primitive

The following table summarizes the options available in the RAM primitive type.

Chip Enables	0, 1, 2 or 3 active-low chip enables. If any chip enable input is high, all read and write functions are disabled. In no enable is provided, the device will always be enabled.
Write Enable	The active-low Write Enable pin is not optional. A low level on this pin causes the data present at the Data In lines to be written to the location selected by the address lines.
Output Enable Pin	This active-low pin controls output enable but does not affect writing.
Data In/Out	The data input and output lines can either be separate or combined into a single I/O bus.
Three-state outputs	If the input and output lines are combined, or if three-state outputs are specified, the outputs enter a high-impedance state if Write Enable or any Chip Enable are high. If three-state outputs are not specified, data outputs will be high when disabled.
Single-word Simulation	If this option is selected, only a single word of real memory will be allocated for simulation purposes, i.e. the address inputs will be ignored. This allows logic testing of a circuit containing a large RAM device without consuming large amounts of program memory.
Common I/O	This option specifies that a single I/O pin will be used per data bit, rather than separate data in and data out lines. In this case, three-state outputs are assumed and outputs will be disabled when writing.

Don't Know Input Handling

If any combination of Don't Know values on the control inputs could cause a write, then the selected memory location is invalidated (i.e. the location will contain Don't Know values). If the address inputs also have Don't Know values, then the entire device will be invalidated.

RAM Pin Options

The normal pin delay (using the Delay.Pin attribute field) and pin inversion (using the Invert.Pin attribute field) options can be used with RAM devices.

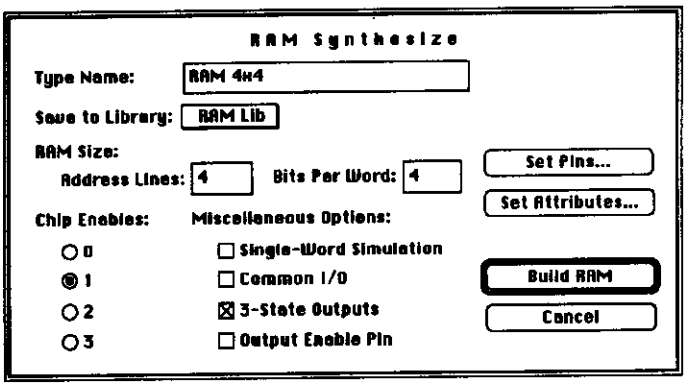
See Chapter III - Simulation for more information on pin delay and inversion.

Creating Custom RAM Devices

The RAM tool allows you to create RAM part entries in a library with any combination of optional features.

NOTE: The RAM tool will require you to select a library to which the new device definition will be saved. You may wish to create a library for this purpose (using the pop-up menu in the parts palette) before proceeding.

Select the RAM item in the Tools menu. The following parameters box will appear:



The following table summarizes the controls in this box.

Type Name	The device will be saved to the library under this name.
Save To Library	This pop-up menu allows you to select an open library to save the device to.
Address Lines	This specifies the number of address lines for the device, in the range 1 to 30
Bits Per Word	This specifies the number of bits (1 to 256) per memory location, which will also be the number of data input pins and data output pins.
Chip Enables	These buttons allow you to select 0, 1, 2 or 3 chip enable inputs.
Single-word Simulation	This check box enables the single-word simulation option, described in the previous section.
Common I/O	This option specifies that a single I/O pin will be used per data bit, rather than separate data in and data out lines.
Output Enable Pin	This will add an additional control pin which controls output enable but does not affect writing.
Set Pins	This button will display a box allowing pin names and default settings to be edited for each pin.
Set Attributes	This button displays the standard attribute edit box for part attributes.
Build RAM	When this button is clicked, a device having the selected features will be created and written to the selected library.

RAM Size Limits

RAM devices must meet all of the following limits:

- 30 address line inputs.
- 256 bits per word.
- Total RAM memory space < 2^{31} bytes (even if single word simulation is used).
- Sufficient program memory free to allocate a block twice the size of the RAM memory space, unless single word simulation is used.

Editing RAM Devices

There is no way to re-enter the RAM parameters box with an existing device. Once a RAM device definition has been created in a library, limited changes can be made by editing it using the DevEditor tool.

> IMPORTANT NOTE: The RAM device definition contains structure information that cannot be edited after the device is created. Adding or deleting any pins using the DevEditor will invalidate the device definition and render it useless.

The DevEditor can be used to make the following changes to a RAM device, if desired:

- any graphical changes to the symbol.
- pin name, visible pin number or pin attribute changes (including pin delay and inversion).
- limited pin type changes (e.g. changing to open collector).
- part attribute changes.

Chapter VIII - Programmable Devices

This chapter provides details on creating and using PROM (Programmable Read Only Memory) and PLA (Programmable Logic Array) devices with user-specified data. The use of the FromABEL tool to read external data files and simulate industry Programmable Logic Device types is covered later in this chapter.

The PROM and PLA Primitive Types

DesignWorks supports the direct simulation of PROM and PLA devices as primitives. This means that these devices can be efficiently represented as a single simulation device, rather than having to generate a netlist of equivalent logic devices.

The PROM and PLA devices represent "raw" memory or PLA (AND-OR) arrays. The primitive device models do not include capability for registers, feedback, three-state buffers or other device features. In order to model these features in industry PLD and PROM types, the FromABEL tool automatically generates a sub-circuit model made up of a raw PLA device, plus other primitive registers, buffers, etc. as needed.

PROM Device Format

For the purposes of simulation in DesignWorks, a PROM (Programmable Read Only Memory) is defined as a device having N inputs (from 1 to 30) and M outputs (from 1 to 256), and having 2^N storage locations, each containing M bits. Each different input combination selects one of the storage locations, the contents of which appear on the output lines. The number of storage locations required doubles for each input bit added, so PROM organization is only practical for a relatively small number of inputs. The advantage of the PROM is that any arbitrary Boolean function can be represented simply by storing the truth table for the function in the appropriate storage locations.

PROM Size Limits

PROM devices must fall within all of the following limits:

- 30 address line inputs.
- 256 bits per word.
- Total PROM memory space < 2^{31} bytes.
- Sufficient program memory free to allocate a block twice the size of the PROM memory space.

PLA Device Format

In DesignWorks, a PLA (Programmable Logic Array) consists (in effect) of a group of AND gates feeding into a single OR (active high) or NOR (active low) gate for each output bit. Each AND-gate input is connected to either an input bit, the inverse of an input bit, or constant high. By selectively making these input connections it is possible to determine which input combinations will produce 0's or 1's in the outputs. PLAs are actually represented internally in a compact binary format, not as a netlist of AND and OR gates.

The input connections required to implement simple logic functions can generally be determined "by eye" for simple cases, whereas more complex logic must be reduced using Karnaugh maps, the Quine-McClusky method or other more advanced design techniques. These methods are discussed in numerous circuit design textbooks and will not be covered here. DesignWorks has the capability of reading device data produced by external logic compiler programs such as MacABEL™.

PLA Size Limits

PLA devices must fall within the following limits:

- Number of Inputs + Number of Outputs <= 800.
- Number of product terms per output <= 65535.

Complex Programmable Logic Devices

The term Programmable Logic Device will be used here to refer to a real programmable device which consists of one or more AND-OR planes plus associated registers, buffers, feedback paths, etc. There is no PLD "primitive" device in DesignWorks. Some very simple PLDs can be directly simulated with a single PLA primitive type, for example a PAL10L8 type device. However, most PLDs are simulated by creating a sub-circuit containing one or more PLA primitive devices plus the other

required logic and wiring. In most cases, these sub-circuits are created automatically by the FromABEL tool, which is described later in this chapter.

Creating PROMs, PLAs and PLDs for Simulation

No manual method of entering PROM or PLA data is provided in DesignWorks. The internal data for these devices must be read from a text file in a specific format. These files are intended to be machine-generated using a program such as MacABEL™, or some other compiler package.

Two general methods are available for creating these devices, depending on type:

- PROM devices are created using the PromPLA tool. This tool allows you to manually specify the format of the device (i.e. number of inputs and outputs) and then to read the internal data from a hex file in standard format.
- PLA/PLD devices are created using the FromABEL tool. All information about the device, including the number and exact functions of inputs and outputs, is described in a text file. This file is known as a DesignWorks Link File (or "dwill", after the ".dwl" file name extension used). These files are normally created using the MacABEL programmable logic compiler, although other external means can be used to produce them.

The formats of these files are described in technical notes provided with this manual.

Creating PROM Devices

The standard hex file format used for loading PROM data does not include any information on device structure, but only a raw listing of binary data. For this reason, the PromPLA tool includes the ability to specify the parameters of a PROM device manually. The internal data must always be read from a file.

Generating the Data File

The hex files used for PROM loading are in the commonly-used Intel MDS hex format. This format can be generated by the MacABEL logic compiler and by many microprocessor assemblers and compilers. The file format is described in a technical note provided with this manual.

The PromPLA Tool

The PromPLA tool allows you to manually specify the format of the device (i.e. number of inputs and outputs) and then to read the internal data from a hex file in standard format.

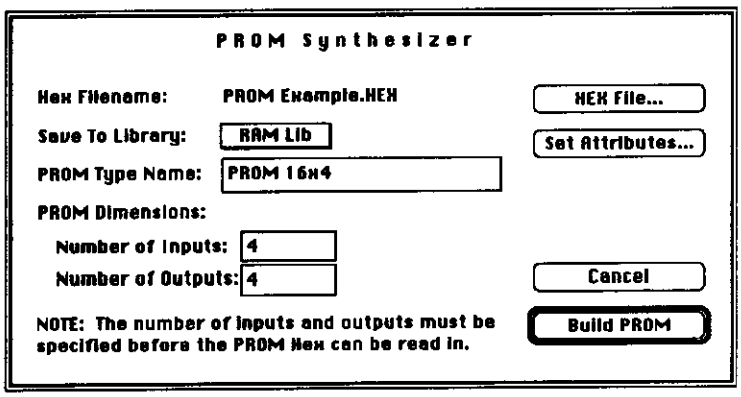
Installation

See Chapter II - Getting Started for general information on installing tools.

Starting PromPLA

NOTE: The PromPLA tool will require you to select a library to which the new device definition will be saved. You may wish to create a library for this purpose (using the pop-up menu in the parts palette) before proceeding.

To invoke PromPLA, select it from the Tools menu. This opens the following parameter box:



The following table summarizes the options in this box.

Hex Filename	This shows the currently-selected input file. This is set using the HEX File button.
Save To Library	This pop-up menu allows you to select the destination library for the completed part.
PROM Type Name	The name of the part as it will appear in the parts palette and on the symbol.
Number of Inputs	The number of address inputs.
Number of Outputs	The number of data outputs, i.e. the number of bits per storage location.

HEX File	This button allows you to select the hex data file. By default, only files with the extension ".hex" are shown. A button on this file box allows you to display all text files. The format of this file is described in a technical note supplied with this manual.
Set Attributes	This button displays the standard attribute edit box allowing you to add default attribute values.

Editing PROM Devices

There is no way to re-enter the PROM parameters box with an existing device. Once a PROM device definition has been created in a library, limited changes can be made by editing it using the DevEditor tool.

IMPORTANT NOTE: The PROM device definition contains structure information that cannot be edited after the device is created. Adding or deleting any pins using the DevEditor will invalidate the device definition and render it useless.

The DevEditor can be used to make the following changes to a PROM device, if desired:

- any graphical changes to the symbol.
- pin name, visible pin number or pin attribute changes (including pin delay and inversion).
- limited pin type changes (e.g. changing to open collector).
- part attribute changes.

Reloading PROM Devices

A PROM can be reloaded from a new hex file after it is placed on the schematic, by the following procedure:

- Select the device on the schematic.
- Select the PromPLA item in the Tools menu.
- Click on the Reload button.
- Select the desired new hex file.

The updated device can be saved back to a library, if desired, using the Schematic tool's Save to Library command.

Viewing PROM or PLA Contents

The contents of an existing PROM device on a schematic can be viewed by selecting the device on the schematic, then selecting the PromPLA item in the Tools menu.

Creating Programmable Logic Devices

The FromABEL Tool

The FromABEL tool allows you to synthesize Programmable Logic Devices (PLDs) designed with MacABEL™ 1.1 or newer software. This is used to supplement the MacABEL software by allowing you to simulate PLDs with the rest of your design in the DesignWorks environment.

Installation

See Chapter II - Getting Started for general information on installing tools.

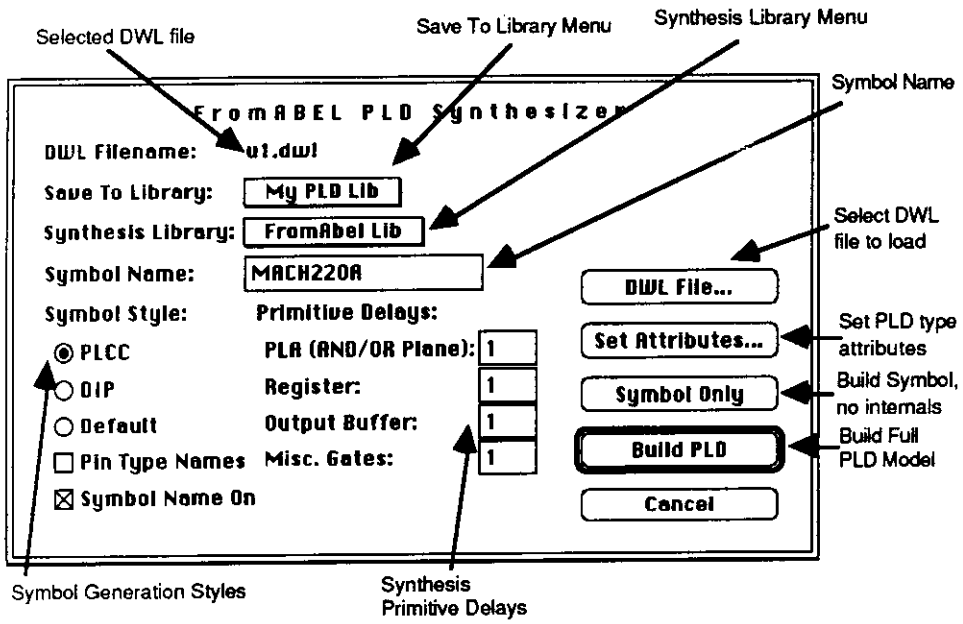
Generating the Data File

The MacABEL software provides an option to allow you to produce a "DesignWorks Link File" (*.dwl). These files describe the structure and logic of the MacABEL PLDs. Refer to MacABEL's Simulate Equations, Simulate Optimized and Simulate JEDEC commands.

Starting FromABEL

To invoke FromABEL, select it from the Tools menu. This opens the parameter dialog described in the following section. Proceed by selecting the desired ".dwl" file and setting any desired synthesis options. Then initiate the FromABEL building process. FromABEL will display a synthesis progress box showing details of the building process.

The FromABEL Parameters Dialog



The following table summarizes the features available in the FromABEL Parameters Dialog.

DWL Filename	This displays the last DesignWorks Link file selected and successfully loaded for synthesis operations. All structural & logical information will be extracted from this file.
Save To Library Menu	This pop-up menu displays the list of DesignWorks libraries currently open. Choose the destination library for the newly built PLD from this list.
Synthesis Library Menu	This pop-up menu displays the list of currently open DesignWorks libraries. Choose the source library for PLD synthesis primitives from this list (usually "FromABEL Lib", supplied with the package).
Symbol Name	Set to the value specified in the last DWL file loaded. Can be edited to a new name. This is used as the library name and the symbol name, if enabled (See "Symbol Name On" below).

PLCC Symbol Style	Creates a symbol with an equal number of pins on each of the four sides. Pin #1 is placed at the top-middle of the symbol and the other pins are added sequentially from this point in a counterclockwise fashion. Note that all unused pins will appear on the symbol. This is useful for large parts where the default method would create very tall symbols.
DIP Symbol Style	Creates a symbol where the number of pins are split equally on the left and right sides. Pin #1 starts in the upper-left and proceeds sequentially down the left side and up the right side. Note that all unused pins will appear on the symbol.
Default Symbol Style	Creates symbol using only the pins declared in your MacABEL source file and places any pin that is an OUTPUT-only signal on the right side and all others on the left side. Pin ordering is dependent on their declarations and spelling in the MacABEL source file.
Pin Type Names	Enabling this option will cause FromABEL to ignore your MacABEL pin names and use standard names for pins based on their I/O types: I - Dedicated input pin, O - Dedicated output pin, IO - Bidirectional pin, GND - Ground pin, VCC - Power pin, NC - Unused pin.
Symbol Name On	Enabling this option will cause FromABEL to generate a symbol with name that is entered in the "Symbol Name" text field, described above.
Primitive Delays	The four primitive delay text fields require an integer value (0-32767). This will allow you to obtain a functional model that more closely models the real device. The default delay for synthesis primitives is one (1). Avoid using values of zero (0), especially where feedback is concerned.
DWL File	Brings up a select file dialog box allowing you to choose the DesignWorks Link file which contains the PLD description.
Set Attributes	Brings up the standard DesignWorks attribute editing dialog. You edit or add attributes that will be specific to this PLD type. See the DesignWorks/Schematic Reference Manual for information on attributes and usage of this dialog.

Symbol Only	This causes FromABEL to ignore the structural internals of your PLD and produce a symbol only for schematic purposes, i.e. no simulation.
Build PLD	This will cause FromABEL to build the entire PLD including the internal circuit for simulation from the DWL file selected and the settings in this dialog.
Cancel Button	Quit FromABEL without doing anything.

The FromABEL Synthesis Process

The FromABEL tool makes use of many DesignWorks resources. It uses the Schematic, LibIO, DevEditor and PromPLA tools via the MEDA interface in creating PLDs. This is done invisibly and in memory, i.e. it doesn't write out any circuit files to disk. The FromABEL synthesis process is as follows:

- Create the main PLA(s) using the PromPLA tool.
- Load the synthesis elements from the library.
- Calculate the pre-place & routing parameters of the internal circuit.
- Place the main PLA(s) in the internal circuit.
- Build and attach the input nodes to the internal circuit.
- Build and attach the output nodes to the internal circuit.
- Route any global signals such as clocks, output enables, etc.
- Build new type using the DevEditor.
- Attach new internal circuit to the new type.
- Auto generate a symbol for the PLD with user settings
- Add necessary DesignWorks attributes to the new PLD for later schematic and netlist purposes.
- Save new PLD type record to the appropriate library.

This process can take anywhere from 5 seconds to a couple of minutes depending on the PLD's complexity and size.

Limited or Unsupported PLDs

The FromABEL tool doesn't guarantee to correctly build or support the following PLDs that MacABEL 4.20 supports:

F507	F839	P16Z8	P22IP6	P23S8
P241	P330	P331	P332	P448

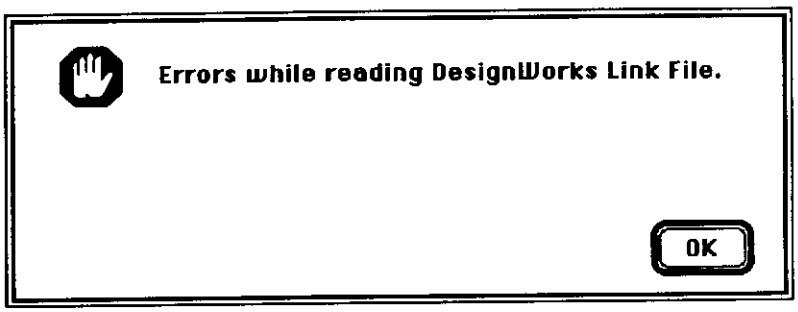
IMPORTANT: If the PLD you are using is NOT supported by FromABEL it may still synthesize without any errors. In this case, the functional simulation results are not guaranteed to be accurate.

FromABEL may correctly build some variations of the E5AC312, P241, P330, P331, and P448 parts.

Limited support of the P16RA8 and P20RA10 is available in registered mode only. Similarly, the P32VX10 has limited support for its bypass mode.

FromABEL Errors

The FromABEL errors box is used to display status messages about why FromABEL failed to create your PLD:



The most common errors are related to not having enough memory to build the PLD. Other errors result from FromABEL not finding the resources it requires to build your PLD, which again can be memory related. The other errors that may be displayed in this box include:

Errors while reading DesignWorks Link File	This will occur generally when there isn't enough memory to load the data in from the ".dwl" file. Alternatively, this can come up if the file has any been corrupted or doesn't match the version support by your copy of FromABEL.
Error trying to open link file	This will occur if the file is busy, i.e. opened in some other application.
Unsupported PLD or function, CODE = 6000	This error message will come if you are using a PLD with such a configuration that is unsupported by the internal circuit building phase of FromABEL. Take note of the "CODE" number that appears and call for MacABEL Technical Support. This error generally occurs when using a PLD that isn't supported by FromABEL.

Chapter IX - Menu Reference

The DesignWorks Simulator is made up of the Simulator core tool, called Sim, and a number of optional tools. Since each tool has its own menu commands, the menu items that are available and enabled depend upon which tools are installed and which window is currently active.

Note re TestPanel Menus

The TestPanel is a special case because it is a "palette" and always floats above the schematic and timing windows. For this reason is has its own pop-up menus in the TestPanel window and does not use any of the menu commands in the menu bar at the top of the screen. The menus at the top of the screen will continue to affect the current document window and do not affect the TestPanel.

TestPanel menus are described in Chapter VI - The TestPanel Tool.

File Menu Commands

The following commands are all associated with the Timing tool. The TestPanel has its own File pop-up menu.

Open Timing Text

This command clears the timing diagram, then opens the selected timing text data file and pastes the data onto the diagram. This is equivalent to selecting the Clear Simulation command, then using the Paste command to place the file data at time zero. See the rules for the Paste command below.

NOTE: This command does not display or remove any traces in the timing diagram. It only reads signal event data and associates it with matching traces. If any traces are named in the file that are not currently displayed, you will be warned and that data will be skipped.

Save Timing Text As

This command saves all the displayed data in the timing diagram to a text file. This file can be used for external purposes, or can be reloaded as a setup for a new simulation using the Open Timing Text command.

> *See Appendix A - Timing Text Data Format for a description of the file format.*

Print Timing

This command prints the contents of the timing window using the current page setup. The current display will be divided into as many pages a required.

Page Setup

This command determines the page setup for the Print Timing command. This can be different than the setup for the schematic diagram.

Edit Menu Commands

The following commands are all associated with the Timing tool. The TestPanel has its own Edit pop-up menu which is described later in this chapter under TestPanel Menus.

Undo

This command undoes the last editing operation in the Timing window. Unlike the Schematic tool, Timing supports only a single Undo and no Redo operation.

Cut

This command copies to the clipboard any signal change events on selected signals in the selected time interval and clears the selected interval. The data is stored in both picture form and text data form (i.e. TestPanel command

format). Events after the selected interval are not moved forward. The Delete Time command can be used to do this.

NOTES:

1) If you wish to paste a timing picture into a word processing package, it may be necessary to paste it first into a drawing program to select the picture data on the clipboard. A word processing package will normally take the text data from the clipboard by default.

2) You cannot modify timing data that is older than the current simulation time.

See Appendix A - Timing Text Data Format for information on the text format used to store timing data on the clipboard.

Copy

The Copy command copies the selected timing data to the clipboard in picture and text format. See the notes under the Cut command, above.

Note that Copy can be used on a selection to the left of (older than) the current simulation time since it does not modify the selected data.

Paste

The Paste command pastes the text timing data from the clipboard onto the selected area of the timing diagram. The following rules are used for matching the data on the clipboard with the selected interval on the timing diagram:

- Data is always pasted by name, i.e. the name of a signal in the clipboard data will be matched the same-named signal in the timing diagram. Neither the order of the signals in the clipboard data or the selected status of traces on the timing diagram is significant. To paste data from one signal to a signal with a different name, it is necessary to paste it first into the TestPanel or into a text editor, modify the names, then paste it back.

  The Paste operation affects only signals named in the clipboard data, regardless of the selection on the timing diagram.

- The Paste operation will not locate signals in the schematic that are not currently displayed in the timing diagram. No new traces will be added by this operation.

- If the time interval selected on the timing diagram is non-zero width then the selected interval is deleted and all later events on pasted signals are moved forward. A time interval equal to the width of the clipboard data is then inserted and the new data pasted into this interval.

See more information on timing diagram editing in Chapter V - The Timing Diagram.

Clear

The Clear command clears any signal change events in the selected area on the timing diagram.

NOTE: This may affect the value of a signal after the selected time interval. E.g. removing a 0-to-1 transition from a signal will leave the signal in a zero state indefinitely.

Duplicate

This command inserts a duplicate of all selected signal data on the timing diagram after the selected interval. I.e. the selected data is copied to a temporary location, then the selection point is moved to the end of the selected interval and the copied data is inserted at this point. All signal changes after the duplicate data are moved back in time by the width of the original selection.

Point

This places the timing diagram in Point cursor mode. This mode can be used to select data for clipboard operations.

Draw Sig

This command places the timing diagram in Draw Signal cursor mode. This mode allows drawing and editing signal change events.

Timing diagram editing is discussed in Chapter V - The Timing Diagram.

Select All

This command selects all traces and the entire time interval of the timing display.

Insert Time

This command inserts a blank time interval in the selected traces. The new interval is inserted in front of the selected interval and is of the same width as the selected interval.

Delete Time

This command deletes the selected time interval from the selected traces and moves all later data ahead by the width of the interval.

Simulation Menu Commands

The Simulation menu contains commands for controlling simulation and timing diagram functions, and setting device simulation characteristics. If the Timing tool is not installed, the commands affecting the timing window will not appear.

Core Simulation Commands

Speed

The Speed sub-menu is used to control the simulation speed, i.e. the amount of delay inserted between simulation steps. Simulation speed can be set individually for each open circuit.

Stop

This command stops the simulation immediately. No simulation processing is done when the simulator is in this state.

Run

This command tells the simulation to proceed as fast as possible.

Other Simulation Speeds

The intermediate speed settings between Stop and Run insert various amounts of delay between executing successive simulation time steps. These can be used to slow the simulation progress for convenient observation.

Single Step

This command simulates one time step. To perform the single step, the simulator looks at the time value associated with the next signal change event in the queue, simulates the effect of that and all following events scheduled at the same time, then returns to the stopped state. The actual time value of a single step depends on the nature of the circuit.

Simulation Params

The Simulation Params command is a general method of setting device and pin delays and options. If no devices or pins are selected in the circuit then Simulation Params will be disabled.

The type of box that is displayed will depend upon the types of devices selected, as described in the following:

Selection	Params Box	Notes
A single CLOCK device	Clock Params Box	Only one clock device can be set at a time
A single SETUPHOLD device	SetupHold Params Box	Only one SetupHold device can be set at a time.
A single ONESHOT device	One Shot Params Box	Only one One Shot device can be set at a time.
Any other combination of one or more devices or pins	General Delay Box	Any selected CLOCK, ONESHOT, SETUPHOLD, sub-circuit or other non-delay devices will be ignored for device delay calculations. Pin delays can be set on these devices.

NOTES:

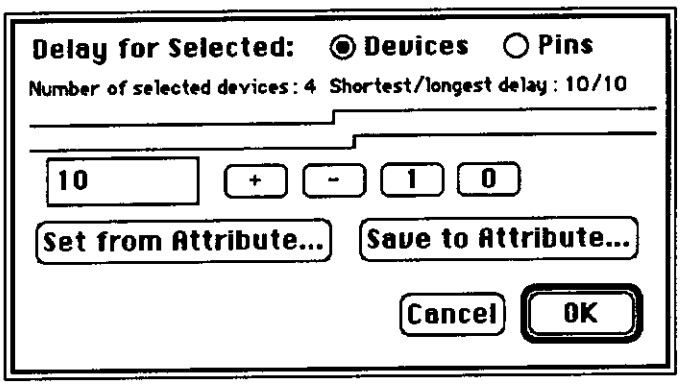
1) You cannot set the device delay of a sub-circuit device since its general delay characteristics are determined by its internal circuit. If any sub-circuit devices are selected, they will be ignored for device delay purposes. You can set the pin delay on sub-circuit devices to modify the path delay through a particular pin. Delays on the devices or pins inside a sub-circuit device are not affected by any settings on the parent device using this command.

2) The Simulation Params command relies on numeric information in a specific format being present in the Delay.Dev or Delay.Pin attribute fields. Any invalid information in these fields will be ignored and default values used.

See more information on the usage of specific attribute fields for simulation parameters in Appendix B - Simulation Attribute Fields

General Delay Box

For any collection of devices and pins with delay characteristics, the following box is displayed:



The controls in this box are summarized in the following table.

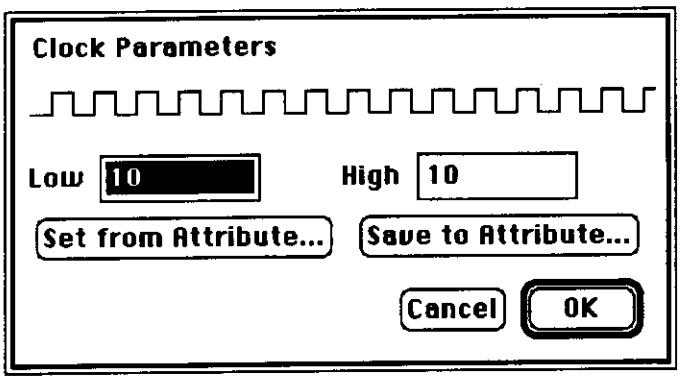
Devices	When this button is enabled, the other controls display and set the *device delay* characteristic of the devices currently selected in the circuit. Items such as CLOCK or sub-circuit devices which have no device delay characteristic will be skipped.
Pins	When this button is enabled, the other controls display and set the *pin delay* characteristic of the pins currently selected in the circuit.

Number of selected devices/pins	This shows a count of the devices or pins that will be affected by changes made in this box.
Shortest/Longest delay	This shows the shortest and longest delays found in any of the selected devices or pins. (Note that each device or pin has only a single integer delay value associated with it.)
Delay Text Box	If all selected devices or pins have the same delay value, it is shown in this box. If a variety of values exist among the selected items, this box will be empty. Typing a new value (between 0 and 32767) in this box will set all items to the given value.
+	Clicking this button will add 1 to the delays in all selected items, to a maximum value of 32767.
-	Clicking this button will subtract 1 from the delays in all selected items, to a minimum value of zero.
1	Clicking this button will set the delay in all selected items to 1.
0	Clicking this button will set the delay in all selected items to zero.
Set From Attribute	Clicking this button will display the Set From Attribute box, allowing the delays in all selected objects to be loaded from a selected field in each object. This box is described below.
Save To Attribute	Clicking this button will display the Save To Attribute box, allowing the delays in all selected objects to be saved to a selected field in each object. This box is described below.

See Chapter III - Simulation for more information on the meaning and usage of device and pin delays.

Clock Params Box

When a single clock device is selected, the following parameters box is displayed:



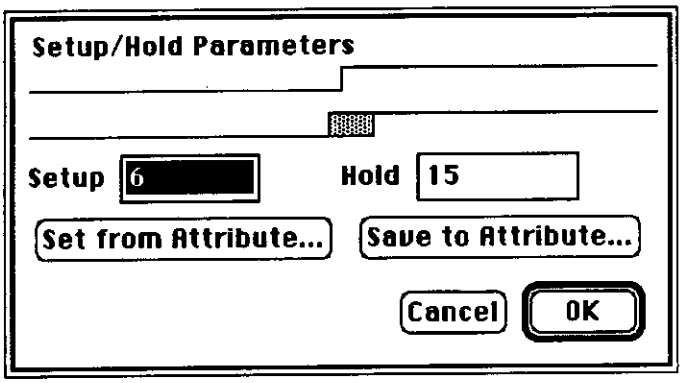
The controls in this box are summarized in the following table.

Low	This text box allows you to edit the low time setting of the selected clock device. Allowable settings are in the range 1 to 32767.
High	This text box allows you to edit the high time setting of the selected clock device. Allowable settings are in the range 1 to 32767.
Set From Attribute	Clicking this button will display the Set From Attribute box, allowing the parameters in the selected clock device to be loaded from a selected field. This box is described below.
Save To Attribute	Clicking this button will display the Save To Attribute box, allowing the clock parameter in the selected device to be saved to a selected field. This box is described below.

See Chapter VII - Primitive Devices for more information on how you can set the startup delay and initial value of a CLOCK device by setting the pin delay and inversion on the output pin.

SetupHold Params Box

When a single SETUPHOLD device is selected, the following parameters box is displayed:



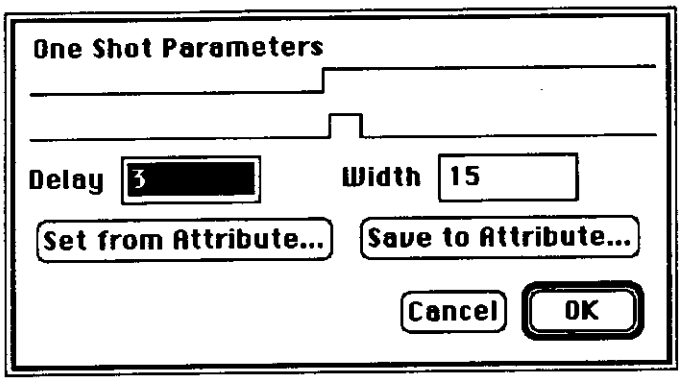
The controls in this box are summarized in the following table.

Setup	This text box allows you to edit the setup time setting of the selected device. Allowable settings are in the range 0 to 32767.
Hold	This text box allows you to edit the hold time setting of the selected clock device. Allowable settings are in the range 0 to 32767.
Set From Attribute	Clicking this button will display the Set From Attribute box, allowing the parameters in the selected device to be loaded from a selected field. This box is described below.
Save To Attribute	Clicking this button will display the Save To Attribute box, allowing the setup and hold parameters in the selected device to be saved to a selected field. This box is described below.

See Chapter VII - Primitive Devices for more information on how you can set the initial value of a SETUPHOLD device by setting the inversion on the output pin.

One Shot Params Box

When a single ONESHOT device is selected, the following parameters box is displayed:



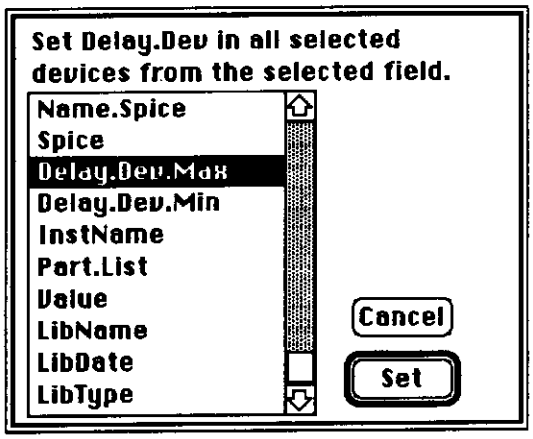
The controls in this box are summarized in the following table.

Delay	This text box allows you to edit the delay time setting of the selected device. Allowable settings are in the range 0 to 32767.
Width	This text box allows you to edit the width time setting of the selected clock device. Allowable settings are in the range 1 to 32767.
Set From Attribute	Clicking this button will display the Set From Attribute box, allowing the parameters in the selected device to be loaded from a selected field. This box is described below.
Save To Attribute	Clicking this button will display the Save To Attribute box, allowing the parameters in the selected device to be saved to a selected field. This box is described below.

See Chapter VII - Primitive Devices for more information on how you can set the initial value of a ONESHOT device by setting the inversion on the output pin.

Loading Delays from Attributes

Each of the simulation parameters boxes described above has Save to Attribute and Set From Attribute buttons. Selecting either of these buttons displays a box like the following (the Set button in this picture will appear as Save if the Save to Attribute option was clicked):



Set From Attribute

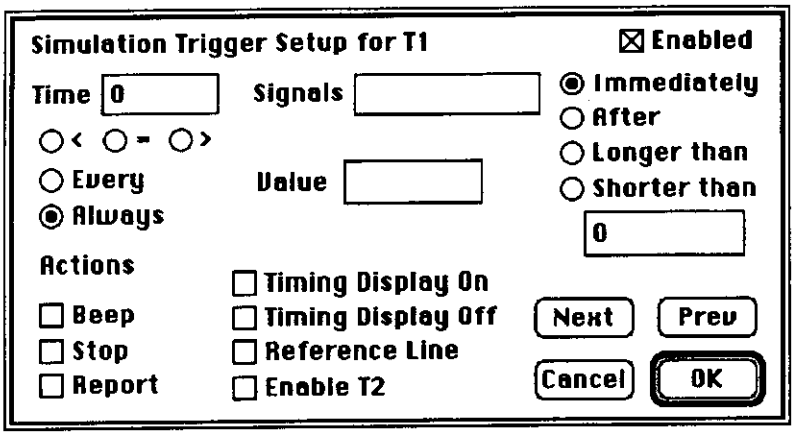
Selecting an attribute field in the list and clicking on the Set button will cause the contents of the selected field to be copied to the Delay.Dev or Delay.Pin field of all selected devices or pins. The copied value will completely replace the old contents. If the selected field is empty in a given object, the Delay.Dev or Delay.Pin field is left unchanged in that object.

Save to Attribute

Selecting an attribute field in the list and clicking on the Save button will cause the contents of the Delay.Dev or Delay.Pin field to be copied to the selected field in all selected devices or pins. The copied value will completely replace the old contents, even if the source value is empty.

Triggers

This command displays the trigger setup box, as follows:



NOTE: The trigger setup box can also be displayed by clicking on the Triggers button on the timing tool palette.

Trigger Conditions

A trigger is activated when three sets of conditions are met:

- The time condition, i.e. the current simulator time value is less than, equal to, greater than, or a multiple of, a given value.
- Signal value condition, i.e. one or more signals are at specified levels.
- The delay condition, i.e. the trigger is activated after a certain delay, or the signal condition exists for greater or less than a specified amount of time.

Trigger Enabling

When the "Enabled" switch is on, the trigger is "armed" and the selected actions will take place as soon as the trigger's conditions are met. If this switch is off, this trigger is disabled until enabled by the previous trigger or by this switch being clicked.

Time Condition Controls

The controls related to the time condition are summarized in the following table.

Time	In this box you enter the time value as a decimal integer. The meaning of this value is determined by the switches below it.
<, =, >	These buttons indicate that the trigger will be activated when the simulation time is less than, equal to, or greater than the given value, respectively.
Every	This time option specifies that the trigger will be activated every time the simulator time equals a multiple of the specified value.
Always	This specifies that the time condition should be considered to be always true. The time value is ignored.

Signal Condition Controls

The controls related to the signal condition are summarized in the following table.

Signals	In this text box, you can type the names of one or more signals whose values will be compared to the hexadecimal integer value typed in the Value box. One or more signals can be entered using the following formats: CLK — The single signal CLK D7..0 — The signals D7 (most significant bit), D6, D5 ... D0 IN1 OUT3 — The signals IN1 and OUT3
Value	In this box you enter the signal comparison value as a hexadecimal integer. This value is converted to binary and compared bit-for-bit with the signals named in the Signals box. The rightmost signal name is compared with the least significant bit of the value, etc.

Delay Condition Controls

The controls related to the time condition are summarized in the following table.

Immediately	This selection effectively disables the delay condition. The delay value entered in the box is ignored.
After	With this option selected, the trigger will be activated the specified amount of time after the time and signal value conditions are met, regardless of whether they continue to be true.
Longer Than	This delay option specifies that the trigger will be activated after the specified delay, as long as the time and signal conditions are still true.
Shorter Than	With this option selected, the trigger is activated if the time and signal conditions are true for less than the specified delay. I.e. Activation occurs when the time and signal conditions cease to be true if they have been true for less than the specified value.
Delay	The delay value is typed into this box as a decimal integer.

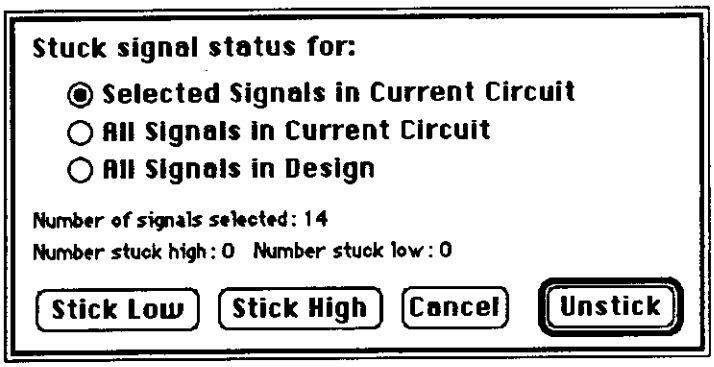
Trigger Actions

When a trigger is activated, any combination of the displayed actions can be invoked.

Beep	Generates a single system beep.
Stop	Stops the simulator immediately.
Display On	Turns on the timing waveform display.
Display Off	Turns off the timing waveform display.
Reference Line	Draws a reference line at this time on the timing waveform display.
Enable T_{N+1}	Enables the next numbered trigger.

Stick Signals

This command allows you to set the "stuck" status of the selected signals. It displays the following box:



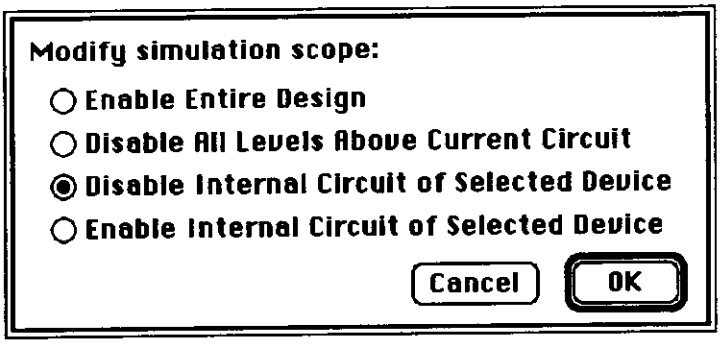
The controls in the box are summarized in the following table.

Selected Signals in Current Circuit	If this option is selected, the Stick High, Stick Low or Unstick option will apply only to signals currently selected in the schematic diagram.
All Signals in Current Circuit	If this option is selected, the Stick High, Stick Low or Unstick option will apply to all signals in the circuit represented in the topmost schematic window, including circuit pages in the same circuit. If this is part of a hierarchical design, only this circuit level is affected.
All Signals in Design	If this option is selected, the Stick High, Stick Low or Unstick option will apply to all signals in all parts of the current design.
Number of signals selected	This displays the number of signals that will be affected by any changes made in this box.
Number stuck high	This displays the number of signals in the selected scope that are currently stuck at a high level.
Number stuck low	This displays the number of signals in the selected scope that are currently stuck low.
Stick Low	This closes the box and applies a "stuck low" value to all selected signals.
Stick High	This closes the box and applies a "stuck high" value to all selected signals.
Unstick	This unsticks all selected signals, allowing them to return to their driven value.

See more information on stuck signal values in Chapter III - Simulation.

Simulation Scope

This command sets the scope of the simulation in a hierarchical design, i.e. to enable or disable parts of the design. The following box will be displayed:



The following table describes the actions available.

Enable Entire Design	Removes all previous disables and enables simulation of all parts of the design.
Disable All Levels Above Current Circuit	The current circuit (i.e. the one displayed in the frontmost circuit window) is set to be the topmost level of the hierarchy that will be simulated. Simulation is disabled in all other hierarchy levels and all driving port connectors at this level will drive at a high impedance level.
Disable Internal Circuit of Selected Device	The internal circuit of the device selected on the schematic will be disabled. All the output pins on the device will be set to a high impedance drive level.
Enable Internal Circuit of Selected Device	The internal circuit of the device selected on the schematic will be re-enabled. Its outputs will revert to the state determined by the internal circuit.

Clear Simulation

Selecting this item clears the timing window (if open), removes all scheduled signal events, sets all devices, signals and pins to their specified initial values (if any) and recalculates output values for all circuit elements.

Clear Unknowns

Selecting this item clears all flip-flop, counter and register primitives to the zero state, and attempts to remove all unknown signal values from the circuit. Note that certain circuit conditions may prevent signals from being placed in a known state:

- unconnected inputs that have not be set to a known level.
- storage devices, such as RAMs that have an unknown stored value.
- any simulation model that does not produce a known output when all inputs are known.

Signal Probe

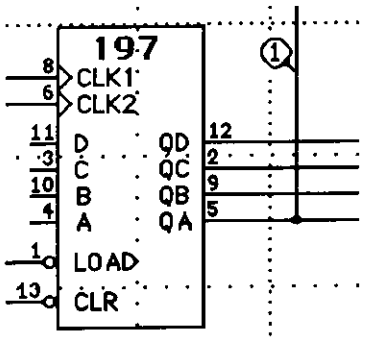
This command places the schematic in Signal Probe mode, allowing you to display and inject signal and pin values. When the probe tip is clicked and held on a signal line or pin, the cursor will show the current value on the signal or pin and track changes that occur as the simulation progresses.

NOTE: The Signal Probe tool can also be selected by the following means:

- Hold the Option key while selecting the Attribute Probe item in the Edit menu, OR clicking on the ? tool on the Schematic tool palette.
- Click the Probe button on the Timing tool palette.

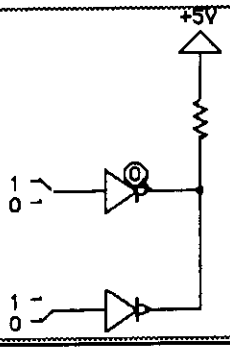
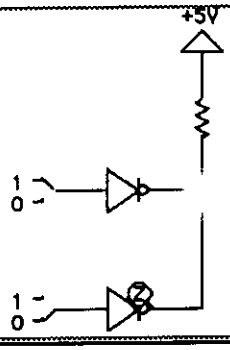
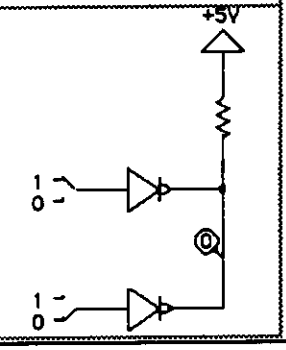
Probing a Signal

Only the signal under the cursor at the time of the click is examined; moving the mouse while the button is pressed does not change the signal being viewed.



Probing a Pin

If the probe tip is clicked on a device pin close to the device body, the probe shows the driving level of that pin, rather than the state of the attached signal. This can be used to resolve drive conflicts in multiple drive situations, as in the following example:

Pin Drive on Upper Device	Pin Drive on Lower Device	Combined Signal Value
		

NOTE: The probe display does not distinguish between low and high drive levels.

Injecting a Value Using the Probe Tool

While the mouse button is held, you can press keys on the keyboard to inject new values onto a signal, as follows:

0	LOW.F
1	HIGH.F
X	DONT01.F
C	CONF.F
Z	HIGHZ
L	LOW.F stuck
H	HIGH.F stuck
space	Unstick

If a stuck value is forced onto a signal, it will not change state until the stuck value is cleared by some user action, regardless of device outputs driving the line. If a non-stuck value is forced, the signal value will revert to its appropriate new level when any change occurs on a device output driving the line.

The spacebar "unstick" command causes the signal to revert to its driven value.

See also the Stick Signals command in this chapter and Chapter III - Simulation for information about stuck values.

Timing Window Commands

Add to Timing

This command adds all selected signals in the current circuit to the timing display. If any selected items are unnamed or they are already displayed, they will be ignored. New items are added at the bottom of the timing display and will be selected after the add.

Add as Group

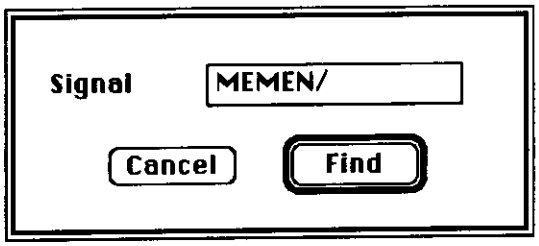
This command is similar to the Add to Timing command except that the selected items are all added as a single group. Where possible, items will be sorted in alphanumeric order with the lowest numbered item in the least significant bit position.

Auto Add

When this item is checked, any signals added or edited on the schematic will automatically be added to the timing window.

Find in Timing

This command allows you to locate a signal or group in the timing display by its label. The following box will be displayed:



The name of any displayed trace can be typed into the Name box. When the Find button is clicked, the item will be located and selected and the window will be scrolled vertically to display it. Note that the search is done by the displayed label in the timing window, not the original name of the item in the schematic.

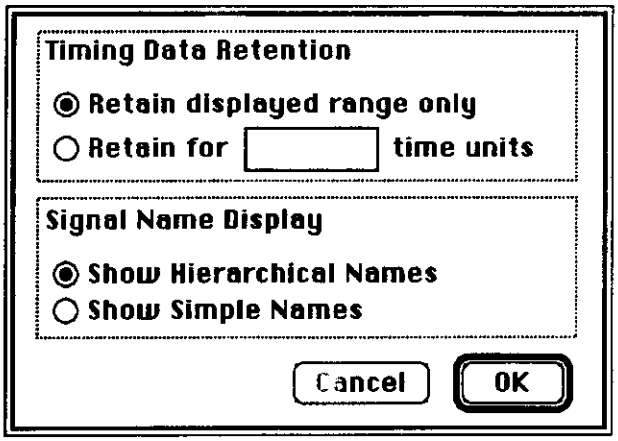
The following "wildcard" characters can be used in this search:

*	Matches any string of zero or more characters. For example "D*" will match "D", "D0", "D12", "DogGone", etc. "*" by itself matches anything.
#	Matches any string of 1 or more digits. E.g. "XA#" will match "XA0", "XA123", etc.
?	Matches any single character. For example, "CTRL?" will match "CTRLX", "CTRL/", etc.

The wildcard characters can be used in combination, as in "DATA#??" which will match "DATA123", "DATA1SQ", etc.

Timing Options

This command displays the following options box:



Timing Data Retention

These options allow you to determine how much signal event data is retained in memory when a simulation is run.

Each time a signal level change occurs DesignWorks creates a record in memory containing a reference to the signal, time, new value and source of the change. In a large simulation these records can consume enormous amounts of memory. This data can be retained for the following purposes:

- for use in refreshing the timing window should it become hidden and then be redisplayed.
- for use in timing window editing operations such as taking the output from one circuit and using it as stimulus for another.

Note that data can be retained only for signals displayed in the Timing window. Signal event data for all other signals is discarded immediately after it is no longer required for simulation.

The option "Retain for Displayed Range Only" is the normal default and results in data being discarded immediately after the corresponding point on the timing display scrolls off the left hand side. This results in minimal memory usage. The setting is equivalent to entering 0 in the retain time box.

The option "Retain for x Time Units" allows you to keep the signal event data for the specified amount of time after it scrolls off the left side of the

screen. If this results in a memory shortage occurring then the simulation will stop and a message will be displayed.

Signal Name Display

This option lets you choose how names of signals in sub-circuits will be displayed on the timing diagram. The option "Show Hierarchical Names" will display the complete "path" to the signal, i.e. prefixed with the names of all parent devices. The option "Show Simple Names" will show only the basic signal name in the label area of the diagram.

Show/Hide Timing Palette

This command displays or removes the timing tool palette. It can be invoked without the timing window displayed to make use of the current time display and simulation controls.

See Chapter V - The Timing Diagram for more information on the timing tool palette.

Timing Diagram Pop-Up Menu

Holding the command (⌘) key pressed while clicking in the label area of the timing diagram will display the timing pop-up menu.

These menu functions control the display of selected signals in the timing diagram and are summarized in the following sections.

Get Info

For groups, this command displays a box allowing reordering of the signals in the group. This affects the way the combined integer value is shown in the timing display.

For individual signals, a signal info box is displayed.

Go To Schematic

This command displays the signal corresponding to this trace in the schematic. If the trace is a group, it shows the corresponding bus, if any, or the first signal in the group.

Remove

This command removes the selected traces from the timing window. THIS CANNOT BE UNDONE!

Group

Combines all the selected traces into a single display group. If any of the selected traces were already grouped, they are in effect Ungrouped first and then recombined with other selected items into a single new group.

Ungroup

Breaks all signals in selected groups into individual traces.

Collect

Brings all selected items together in the display underneath the topmost selected item. This is used to bring associated signals closer together for easier comparison of timing, etc.

To Top

Sends all selected traces to the top of the timing diagram.

To Bottom

Sends all selected traces to the bottom of the timing diagram.

Appendix A - Timing Text Data Format

When a Copy or Cut operation is done on a selected area in the Timing window, two types of data are placed on the system clipboard:

- A picture of the selected area, in standard PICT format.
- A text description of the signal value changes occurring in the selected area.

This Appendix describes this text data format.

General Description of Format

The clipboard text format used by the Timing tool is actually a subset of the command structure supported by the TestPanel tool. Thus, it is always possible to paste timing data into the TestPanel. Copying data from the TestPanel tool to the timing diagram is also possible with some restrictions. The Timing tool does not support control commands like $REPEAT/$END which are available in the TestPanel.

NOTE: The TestPanel's Copy to Timing command is an alternate way of moving TestPanel data to the timing window. This bypasses the clipboard and does support control commands.

Format Specifics

The following rules describe the timing text data format:

- The data is pure ASCII text with no special binary codes except for standard tab and carriage return characters.
- The format of the data is based on the common "spreadsheet" text data format, i.e. each text item is followed by a tab character, except for the last one on a line which is followed by a carriage return.
- Every line has the same number of text items on it.
- The first line of the text (i.e. up to the first carriage return) is a "header" which indicates the meaning of the items on the following lines, by position.
- The lines following the header are signal value lines. Each line represents one time step. A complete data line is written out each time any value on the line changes. No line is written out for time steps in which none of the represented signals changed value.

Header Format

The header consists of a series of "commands", each starting with a $, which describe the meaning of the corresponding data items on the following lines.

The header always contains the command "$T" (denoting a time column), followed by a tab character, followed by "$D" (denoting a delay column). The remaining items depend on the traces that were selected in the timing diagram.

NOTE: The Timing tool always places the time and delay items in the order given here, although it will accept data with them in any order, or even completely missing. Since time and delay are redundant, either one is sufficient. If they are both missing a default delay value will be used.

Single Signal Items

An individual signal is specified by the text $I (for input) followed by a space, followed by the name of the signal. If the signal contains any blanks or control characters, it will be enclosed in quotation marks.

Grouped Items

Grouped items are denoted by the text "$I" followed by a blank, followed by the name of the group, followed immediately (without any spaces) by a list of the signals in the group, contained in square brackets. Any group or signal name which contains blanks or control characters will be enclosed in quotation marks.

NOTE: The TestPanel tool does not use group names, but will ignore any text immediately preceding the "[]" grouping brackets. For this reason, there must not be a space between the group name and the "[" or the group name will be interpreted by the TestPanel as a signal name.

Data Line Format

Each line following the header must contain one data item for each item in the header line. Thus, the first two items will always be:

- The time at which the events on this line take place. The Timing tool places the absolute time at which the events occurred (i.e. corresponding to the time scale on the diagram) in this column. However, when the data is pasted, the times are considered to be relative to the time of the first data line. This is a decimal integer which may take on any 32-bit unsigned value.

• The delay from this step to the next step. This is redundant information, since it can be derived from the times in the first column. It is provided for compatibility with TestPanel and for improved flexibility in exporting to outside software systems.

NOTES:

1) If the delay and time columns do not match, the longest time is used.

2) The delay on the last line has special significance because it indicates the delay from the last signal change to the end of the selected interval. When pasting, this value is used to determine how much time to insert.

The following items on a line will be signal or group values matching the items in the header.

• For groups or individual signals, the special values ".X" (for DONT01) and ".Z" (for HIGHZ) indicate that all signals covered by this data item are in the given state.

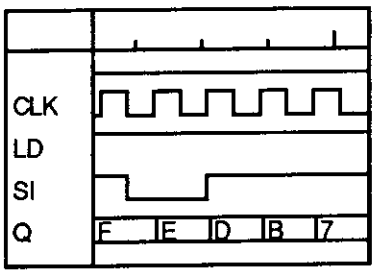
• Individual signals not in Don't Know or High Impedance states will be either "0" or "1".

• Grouped signals which are not all unknown or all high impedance will be specified by a hexadecimal value. The least significant bit of the value corresponds to the rightmost signal in the group list. Special character "X" may be substituted for a hex digit if any of the four signals represented by that digit is unknown, or "Z" if all the signals represented by that digit were high impedance.

Timing Text Example

Following is an example of the timing text data and the corresponding timing diagram.

$T	$D	$I Q[Q0 Q1 Q2 Q3]	$I SI	$I LD	$I CLK
87410	2	F	1	0	0
87412	10	F	1	0	1
87422	10	F	0	0	0
87432	1	F	0	0	1
87433	9	E	0	0	1
87442	10	E	0	0	0
87452	1	E	1	0	1
87453	9	D	1	0	1
87462	10	D	1	0	0
87472	1	D	1	0	1
87473	9	B	1	0	1
87482	10	B	1	0	0
87492	1	B	1	0	1
87493	9	7	1	0	1
87502	4	7	1	0	0



Appendix B - Simulation Attribute Fields

The tools making up the Simulator package make use of a number of attribute fields to store device, signal and pin parameters and simulation setup information. The following table summarizes the usage of these fields.

The following information is provided in the attribute table:

Field Name	The name of the field as it appears in the Define Attributes box.
Type	The type of objects this field is used in.
Tool	The name of the tool that uses this field. If this is blank, then the field is provided as a convenience for the user but not "hard-wired" to any program function.
Inst/Def	Specifies whether the field data is kept with the definition or instance of a sub-circuit. I.e. Instance fields can have a different value in each copy of a sub-circuit.
Description	How the field is used.

Field Name	Type	Tool	Inst/Def	Description
ABELSrcName	Device	FromABEL	Def	Contains the name of the MacABEL source file used to generate this part definition.
Delay.Dev	Device	Sim	Def	Specifies device delay. For most devices, a single decimal integer 0 to 32,767. For Clock, One Shot and SetupHold devices, two integers separated by commas. Should be set using the Simulation Params command and not edited manually.
Delay.Dev.Max	Device		Def	Maximum device delay. This is provided for use with the Load From Attribute button in the Simulation Params command. Not used internally.
Delay.Dev.Min	Device		Def	Minimum device delay. See Delay.Dev.Max.

Delay.Dev.Typ	Device		Def	Typical device delay. See Delay.Dev.Max.
Delay.Pin	Pin	Sim	Inst	A decimal integer specifying pin delay in the range 0 to 32767. Should be set using the Simulation Params command and not edited manually.
Delay.Pin.Max	Pin		Def	Maximum pin delay. See Delay.Dev.Max.
Delay.Pin.Min	Pin		Def	Minimum pin delay. See Delay.Dev.Max.
Delay.Pin.Typ	Pin		Def	Typical pin delay. See Delay.Dev.Max.
DWLSrcName	Device	FromABEL	Def	Contains the name of the ".dwl" file used to generate this part definition.
DWLSrcPath	Device	FromABEL	Def	Contains the "path name" of the ".dwl" file used to generate this part definition, i.e. names of all nested folders, separated by ":".
ExtCctDate	Device	SimLoad/ Schematic	Def	The "last modified" date of the external circuit file, stored as an integer in 1/60 sec since Jan. 1, 1904. Should not be edited manually. See ExtCctName.
ExtCctName	Device	SimLoad/ Schematic	Def	The file name of the external circuit file. Should normally be set using the SimLoad tool or Attach External Sub-circuit command.
ExtCctPath	Device	SimLoad/ Schematic	Def	The "path name" of the external circuit file, i.e. names of all nested folders, separated by ":". See ExtCctName.
Initial.Pin	Pin	Sim	Inst	The initial value for the pin on Clear Simulation. Output pins only. One character: 0, 1, X or Z.
Initial.Sig	Signal	Sim	Inst	The initial value for the signal on Clear Simulation. One character: 0, 1, X or Z.
Invert.Pin	Pin	Sim	Def	Any non-empty value indicates pin inversion.

Sim.InputMap	Design	Sim	Def	Used to specify alternate input mapping for primitive devices. Must be exactly five characters, each one 0, 1 or X. First character specifies value to substitute for low input value, second is high, third is high impedance, fourth is Don't Know, fifth is Conflict. E.g. "011XX" will substitute a high value for high impedance. Default is "01XXX".
Sim.TrigSave	Design	Sim	Def	Used by the Simulator to store the current trigger setup. Should not be edited manually.
TestVectors.Cct	Design		Def	Used to store sets of test vectors for the design. Not used internally.
TestVectors.Dev	Device	TestPanel	Def	Used to store test vectors for the device. Loaded and save by the Place and Test Device and Save Test to Device commands in TestPanel.
Timing.Save	Design	Timing	Def	Used to save the current status of the timing window. Should not be edited manually.

Appendix C - Device Pin Types

Every device pin has a characteristic known as its *pin type*. The pin type is set when the part entry in the library is created and cannot be changed for individual device pins on the schematic.

Refer to the DesignWorks/Schematic User's Manual for information on how to set the pin type while creating a device symbol.

What Pin Types are Used For

For many general schematic editing purposes, the pin type will be unimportant an can be ignored. However, pin type settings are important in the following cases:

- The pin type of each pin is used by the simulator to select what type of output values are generated by a pin. E.g. An open collector output will not generate a HIGH drive level.
- Pin type information is required in many netlist file formats for FPGA layout and digital simulation.
- Correct pin type settings allow the ErrorFind tool to check for fanout and multiple drive situations.
- Other future analysis tools may use this information for timing and loading analysis.

Pin Types Table

The following table lists the function of each of the pin types available in DesignWorks. The Output Value Mapping column specifies how output values specified by the model are mapped to actual pin drive values.

Pin Type	Description	Initial Value	Output Value Mapping
IN	Input - this is the default for pins created using the DevEditor tool. This setting is used for all pins on discretes except those with some digital function. No output value can be placed on an input pin.	HIGHZ	No output drive allowed.
OUT	Output - always enabled.	DONT01	None

3STATE	Output - can be disabled (i.e. high-Z). Note: The three-state capability only exists for specific primitive types that have a three-state enable pin. For other types this will behave like OUT.	DONT01	None
BIDIR	Bidirectional	DONT01	None
OC	Open collector output - i.e. pulls down but not up	DONT0Z	HIGH maps to HIGHZ.
BUS	Bus pin - This does not represent a physical signal but is a graphical representation of a group of internal pins, each having its own type. Bus pins cannot have values and are not supported on primitive device types.	None	None.
LOW	Output - always driving low	LOW	All values converted to LOW.
HIGH	Output - always driving high	HIGH	All values converted to HIGH.
LTCHIN	Input to a transparent latch - this is used for calculating cumulative setup and hold times.	HIGHZ	No output drive allowed.
LTCHOUT	Output from a transparent latch - this is used for calculating cumulative setup and hold times.	DONT01	Same as OUT.
CLKIN	Input to an edge-triggered latch - this is used for calculating cumulative setup and hold times.	HIGHZ	No output drive allowed.
CLKOUT	Output from an edge-triggered latch - this is used for calculating cumulative setup and hold times.	DONT01	Same as OUT.
CLOCK	Clock input - this is used for calculating cumulative setup and hold times.	HIGHZ	No output drive allowed.
OE	Open emitter output - i.e. can pull up but not down.	DONT1Z	LOW maps to HIGHZ.
NC	A no-connect pin	HIGHZ	No output drive allowed.

Device Pin Type and Simulator Efficiency

Incorrect device pin type settings can have a major impact on simulation speed, even in cases where they wouldn't affect the correctness of the results.

Bidirectional Pins

The use of bidirectional pins should be avoided unless specifically required by circuit logic. On primitive types, any value change on a signal attached to a bidirectional pin will cause the device model to be called to reevaluate the device. On sub-circuit devices, the simulator must make several passes through all circuit levels which may affect the value of the signal or be affected by it. Setting a pin on a sub-circuit device to be an input or output greatly reduces this overhead.

Output Pins

If a device pin will only ever be used to drive the attached signal and the device cannot be affected by changes in value on the pin, then it should be an output type. Changes in the value of a signal attached to an output pin do not cause the device model to be called for reevaluation. This is particularly significant for sub-circuit devices.

Input Pins

Pins with an input type setting can never place a drive value on the attached signal. On sub-circuit devices this provides an important hint to the simulator that internal value changes on the attached signal will not affect any other circuit level.

Appendix D - Primitive Device Pin Summary

In DesignWorks primitive device types, the function of each pin is determined by its type (i.e. input or output) and its sequential position in the device's pin list (i.e. as seen when the part is opened in the DevEditor). Pin name is not significant. Each type has specific rules about the ordering of pins. Failure to adhere to these rules will result in incorrect simulator operation.

NOTE: Bus pins are not supported on primitive device types.

For many primitive types, certain control inputs and outputs can be omitted to create simplified device types. E.g. On flip-flop types, the Set and Reset inputs can be omitted. The table below shows which combinations of inputs are allowable and the required order.

Pin Inversion

In addition to the pin function options shown in the table, any pin on any device can be inverted by specifying a value in the Invert.Pin attribute field. Any non-empty value will cause the pin logic to be inverted.

Pin Function Table

The following table lists the pin functions and orders for all primitive device types. In some cases, a number of pins can be optionally omitted, so rules are given rather than enumerating all possible combinations.

See Chapter VII - Primitive Devices for more information.

Primitive Type	Limitations	Pin Names and Types	Possible Pin Orders
NOT	Exactly 1 input and 1 output	IN - in OUT - out	1) IN OUT
AND, NAND, OR, NOR, XOR, XNOR	N inputs: $1 \leq N < 799$	$IN_0..IN_{N-1}$ - in OUT - out	1) $IN_0..IN_{N-1}$ OUT

X-Gate	Exactly 2 ports and 1 enable	X1 X2 - bidir EN - in	1) X1 EN X2
Buffer	N data inputs N data outputs $1 \le N < 256$	$IN_0..IN_{N-1}$ - in $OUT_0..OUT_{N-1}$ - out EN - in	1) $IN_0..IN_{N-1}$ $OUT_0..OUT_{N-1}$ EN 2) $IN_0..IN_{N-1}$ $OUT_0..OUT_{N-1}$
Resistor	Exactly 2 pins	X1 X2 - bidir	1) X1 X2
Multiplexer	L select inputs M output bits $1 \le M \le 256$ N inputs/ output $2^{L-1} < N \le 2^L$ i.e. the number of inputs per output bit can be less than the number of select input combinations	$S_0..S_{L-1}$ - in $IN_{0,0}..IN_{N-1,M-1}$ - in * EN - in § $OUT_0..OUT_{M-1}$ - out * $IN_{n,m}$ is the input routed to output m when select value is n. § An enable input can exist only if $N == 2^L$, otherwise the extra input is assumed to be a data input.	1) $IN_{0,0}..IN_{0,M-1}$ $IN_{1,0}..IN_{1,M-1}$ .. $IN_{N-1,0}..IN_{N-1,M-1}$ $S_0..S_{L-1}$ $OUT_0..OUT_{M-1}$ 2) $IN_{0,0}..IN_{0,M-1}$ $IN_{1,0}..IN_{1,M-1}$ .. $IN_{0,0}..IN_{N-1,M-1}$ $S_0..S_{L-1}$ EN $OUT_0..OUT_{M-1}$ § § Option 2 only if $N == 2^L$
Decoder	L select inputs M output bits $1 \le M \le 256$ AND $2^{L-1} < M \le 2^L$ i.e. the number of output bits can be less than the number of select input combinations	$S_0..S_{L-1}$ - in EN - in $OUT_0..OUT_{M-1}$ - out	1) $OUT_0..OUT_{M-1}$ $S_0..S_{L-1}$ 2) $OUT_0..OUT_{M-1}$ $S_0..S_{L-1}$ EN
Adder, Subtracter	N output bits N "A" operand inputs required N "B" operand inputs optional $1 \le N \le 256$	$A_0..A_{N-1}$ - in $B_0..B_{N-1}$ - in CIN - in $SUM_0..SUM_{N-1}$ - out COUT - out	1) $A_0..A_{N-1}$ $B_0..B_{N-1}$ $SUM_0..SUM_{N-1}$ CIN COUT * * $B_0..B_{N-1}$ CIN & COUT can be omitted in any combination

D Flip-Flop, D Latch	Must have at least D and CLK inputs and Q output	S - set in D - D in C - clock in R - reset in Q - out NQ - inverted out	1) S D C R Q NQ 2) S D C R Q 3) D C R Q NQ 4) D C R Q 5) D C Q NQ 6) D C Q
JK flip-flop	Must have at least CLK input and Q output	S - set in J - J in K - K in C - clock in R - reset in Q - out NQ - inverted out	1) S J C K R Q NQ § 2) S T C R Q NQ *§ 3) T C R Q NQ *§ 4) C R Q NQ § 5) C Q NQ § § NQ can always be omitted * T = J & K tied together
Register	N output bits N input bits $1 \le N \le 256$	$IN_0..IN_{N-1}$ - in CLK - in CLR - in $OUT_0..OUT_{N-1}$ - out	1) $IN_0..IN_{N-1}$ $OUT_0..OUT_{N-1}$ CLK CLR 2) $IN_0..IN_{N-1}$ $OUT_0..OUT_{N-1}$ CLK
Counter	N output bits N input bits (optional) $1 \le N \le 256$	$IN_0..IN_{N-1}$ - in CLK - in LD - in CLR - in UP - in EN - in $OUT_0..OUT_{N-1}$ - out COUT - out	1) $IN_0..IN_{N-1}$ $OUT_0..OUT_{N-1}$ CLK LD CLR UP EN COUT 2) $IN_0..IN_{N-1}$ $OUT_0..OUT_{N-1}$ CLK LD CLR UP COUT 3) $IN_0..IN_{N-1}$ $OUT_0..OUT_{N-1}$ CLK LD CLR COUT 4) $IN_0..IN_{N-1}$ $OUT_0..OUT_{N-1}$ CLK LD COUT 5) $IN_0..IN_{N-1}$ $OUT_0..OUT_{N-1}$ CLK COUT Note: $IN_0..IN_{N-1}$ & COUT can always be omitted
Shift Register	N output bits N input bits $1 \le N \le 256$	$IN_0..IN_{N-1}$ - in CLK - in LD - in CIN - in $OUT_0..OUT_{N-1}$ - out	1) $IN_0..IN_{N-1}$ $OUT_0..OUT_{N-1}$ CLK LD CIN 2) $OUT_0..OUT_{N-1}$ CLK CIN
One Shot		CLK - in CLR - in Q - out NQ - out	1) CLK CLR Q NQ 2) CLK CLR Q

Clock Osc	Exactly one output pin	CLK - bidir	1) CLK
Binary Switch	Exactly one pin	SW - bidir	1) SW
SPST Switch	Exactly 2 pins	X1 X2 - bidir	1) X1 X2
SPDT Switch	Exactly 3 pins	X1 X2 COM - bidir	1) X1 X2 COM
Probe	Exactly 1 pin	PR	1) PR
Hex Keyboard	4 or 5 pins	$X_0..X_3$ - bidir STROBE - out	1) $X_0..X_3$ STROBE 2) $X_0..X_3$
Hex Display	Exactly 4 pins	$X_0..X_3$ - in	1) $X_0..X_3$
SetupHold	Exactly 3 pins	CLK - in DATA - in Q - out	1) CLK DATA Q
Unknown Detector	Exactly 2 pins	D - in Q - out	1) D Q

Credits

DesignWorks, LogicWorks and MEDA are trademarks of Capilano Computing Systems Ltd.

The DesignWorks Simulator were designed by Chris Dewhurst and coded by Chris Dewhurst and Addam Smith of Capilano Computing Systems Ltd.

This manual was written by Chris Dewhurst using Microsoft Word (a trademark of Microsoft Corporation of Bellevue, Washington).

Macintosh is a trademark of Apple Computer, Inc. of Cupertino, California.

MacABEL is a trademark of Data I/O Corporation of Redmond, Washington.

Any problems or questions can be referred to:

Technical Support,
Capilano Computing Systems Ltd.,
406 - 960 Quayside Drive,
New Westminster, B. C.
Canada, V3M 6G2
(604) 522-6200
Fax (604) 522-3972

or on AppleLink:

D0816

This page is intentionally left blank.

Index

Page number in bold print indicates primary reference to that topic.

$DELAY command 50, **55**, **56**, 64, 134
  example 71, 136
$EXPECTED command 50, **53**, 63, 64
  example 70
$INPUTS command 50, **51**, 134
  example 56, 70, 72, 136
$OUTPUTS command 50
  example 56
$REPEAT command **56**, 58, 66, 133
  example 72
$STOP command 56
$TIME command 50, **54**, 65, 134
$WAIT command 50, **54**, 58
  example 71
++ operator 52
  example 56
-- operator 53
.C value code 51, **52**, 53, 63
.DC value code 54
  example 71
.dwl file **101**, 104, 105, 106, 108
.N value code **52**, 53, 63, 70
  example 70
.X value code
  TestPanel **52**, 53
  Timing text data 135
.Z value code
  TestPanel 52, 53
  Timing text data 135
<> zoom option 37
>< zoom option 37
ABELSrcName attribute field 137
Add As Group command 35, **128**
Add To Timing command 11, 34, 35, **128**
adder device **82**, 144
AND device **75**, 143
attributes
  delay 14, 119
  loading CLOCK parameters from 117
  loading delay from 116
  loading ONESHOT parameters from 119
  loading SETUPHOLD parameters from 118
  loading test vectors from 58, 67
  pin inversion 23, 76, 143
  PLDs 106
  PROM 103
  RAM 96, 97, 98
  saving CLOCK parameters to 117
  saving delay to 116
  saving ONESHOT parameters to 119
  saving SETUPHOLD parameters to 118
  saving test vectors to 57, 68
  setting in DevEditor 76
  table 137
  value initialization 8, 18, 19
Auto Add command **129**
bidirectional pins 141
  efficiency 23, 142
  initial values 20
  on sub-circuit devices 22, 23
  PLDs 106
  transmission gate 77
binary switch device **91**, 146
Boolean formulas 99
buffer device 13, **77**, 144
  non-inverting 78
  PLDs 99, 100
buffering sub-circuit pins 15, 22
bus pins 12
busses 11
  bus pins 12, 141
  open collector 9
  Timing 132
capacitance 6, 11, 78
Clear command
  TestPanel 60
  Timing 37, 40, 112
Clear Sheet command 66

Clear Simulation command 18, 37, 84, 89, 109, **126**, 138
Clear Unknowns command 8, 19, 37, 84, **126**
Clear X option 37
clipboard
  copying Timing to TestPanel 58
  TestPanel data format 59
  TestPanel edit commands 48, 59
  Timing edit commands 40, 110
  Timing picture 41
  Timing text format 133
clock device 5, 12, 18, 46, **89**, 146
  delay attribute format 14
  setting parameters 90, 114, 117
  synchronizing 89
clock pulse
  TestPanel 51, 52, 53, 63, 70
clocked devices 14, 15, 70, 83, 85, 93
Collect command 34, 132
conflict logic state 7, **9**, 11, 15, 19, 42, 75, 92, 93, 127
Copy command
  TestPanel 59
  Timing 37, 40, 111
Copy To Timing command 44, 45, 64, **66**, 133
counter device 14, 19, **87**, 126, 145
Cut command
  TestPanel 59
  Timing 37, 40, 110
D flip-flop device 14, 19, 71, **83**, 145
D latch device **84**, 145
decoder device **81**, 144
decrementer device 83
Default Header command **64**, 69
delay **12**
  attributes 14
  clock pin 90
  devices 115
  effect of zero delay 13
  pins **12**, 115
  PLDs 106
  primitive devices 12
  setting 13, 115
  sub-circuit devices 12
  time units 6
Delay.Dev attribute field **14**, 115, 120, 137
Delay.Dev.Max attribute field 14, 137
Delay.Dev.Min attribute field 14, 137
Delay.Dev.Typ attribute field 14, 138
Delay.Pin attribute field **14**, 23, 24, 96, 115, 120, 138
Delay.Pin.Max attribute field 14, 138
Delay.Pin.Min attribute field 14, 138
Delay.Pin.Typ attribute field 14, 138
Delete Cols command 61
Delete Rows command 61
Delete Time command 37, 41, 111, **113**
DesignWorks Link file 101
DevEditor
  attributes 76
DevEditor tool
  attributes 24
  inverted pins 17, 76, 77
  pin order 73, 76, 143
  pin type 22, 140
  PLDs 107
  PROM devices 103
  RAM devices 98
devices
  delay **12**
    setting 114
  effect of unknown inputs 75
  gates 75
  high impedance inputs 15
  initial value 20
  input value mapping 15
  input values 15
  pin delay 12
  pin inversion 17
  pin type 16, **140**
  primitive types 12, 25, **73**, 143
  setup and hold times 15, 53, 70, 83, 93, 118, 141
  simulation models 25
  simulation pseudo-devices 26
  sub-circuit devices 22
discrete components
  pin type 140
Disp On/Off option 18, 37, 123

Don't Know logic state **8**, 15, 19, 42, 75, 92, 135
input value mapping 139
RAM 96
TestPanel 52, 53
Draw Sig command 112
Duplicate command 37, 41, **112**
DWLSrcName attribute field 138
DWLSrcPath attribute field 138
events **5**, 6, 13, 114
clearing 18, 37, 126
editing 40, 66, 110, 112
retaining 130
saving to file 57, 109, **110**
settling 46, 62
TestPanel 44
text format 134
ExtCctDate attribute field 30, 138
ExtCctName attribute field 28, 138
ExtCctPath attribute field 138
external references 27
fall time 5
fanout 140
feedback 5, 13, 37, 99, 100, 106
file formats
PLD (dwl) files 101
timing text format 133
Fill Down command **60**, 70
Fill Right command 60
Find in Timing command **129**
flip-flops
D-type 83
Initializing 84
JK-type 85
setup and hold times 83
forcing logic states 6
FromABEL tool 3, 99, **104**
errors 108
unsupported devices 107
gates **75**, 143
Get Info command 35, **131**
Go To Schematic command 132
ground and power connections 15, 24
PLDs 106
Group command 132
groups **35**
adding to display 128
bit order 35
changing 35
busses 35
creating 35, 132
drawing tools 41
find by name 129
group name 35
ungroup 35, 132
header
Default Header command 64
TestPanel 44, 48, 52
Timing text data 133
hex display device **93**, 146
hex keyboard device **92**, 146
Hide Timing Palette command 36, **131**
hierarchy **21**
attributes 137
delay 12
editing an open circuit 21
hierarchy mode 21
Link to Circuit command 61
pin delays 23
pin type 22
port connector pin type 23
port interface 22
signal names 131
simulation scope 125
stuck signals 124
TestPanel 45
timing window 33
high impedance 7, 15, 24, 75, 92, 125
drawing on timing diagram 42
in switches 78, 91
in Timing text data 135
input value mapping 15, 139
TestPanel 52, 53
hold time (See setup and hold times)
incrementer device 82
Initial.Pin attribute field 18, 19, **20**, 84, 138
Initial.Sig attribute field 8, 18, 19, **20**
Insert Cols command 61
Insert Rows command 60
Insert Time command 37, 41, 113
installation 3

inversion **17**, 76, 143
Invert.Pin attribute field **17**, 23, 90, 96, 138, 143
Invert.Pin field 76
JK flip-flop device 14, 15, 19, **85**, 145
Karnaugh maps 100
Link to Circuit command 45, **61**
Load From Design Attr command 58
logic probe (See probe)
MacABEL 100, **104**
MEDA 5, 25, 107, 147
memory device See RAM or PROM
memory usage 6, 29, 73, 77, 130
  PROM 100
  RAM 95, 97
menus 109
  TestPanel 57
  Timing pop-up 131
models
  attaching 26
  creating 25, 31
  installation 4
  MEDA 25
  purging 30
Monitor Mode command 45, 57, **66**
multiple drive 10
multiplexer device **79**, 144
names
  external model file names 30
  port connector 32
  special names 0 and 1 15
NAND device **75**, 143
nanoseconds 6
NOR device **75**, 143
NOT device **75**, 143
One Shot device **91**, 114, 145
  setting 91, 119
open collector pins 7, 9, 10, 23, 78, 98, 103, 140, 141
open emitter pins 10, 23, 78, 141
Open Test File command **57**, 69
Open Timing Text command 109
OR device **75**, 143
oscillator 89
Page Setup command 110
Paste command
  TestPanel 58, **59**
  Timing 37, 40, 109, 111
pins
  bidirectional 142
  bus pins 141
  delay 12, 14, 22, 23, 83, 90
    on clock devices 89
    setting 114
  efficiency 141
  function table 143
  hierarchical block 23
  high impedance inputs 15
  initial values 19, **20**
  input 140, 142
  input value mapping 15
  inversion **17**, 76, 143
  order in primitive devices 143
  output 140, 142
  output value mapping 140
  pin type 16, **140**
PLA device **99**, 100
  size limits 100
Place and Test Device command **67**, 68, 139
PLCC package type 106
Point command 38, 112
port connector 12, 22, 24, **31**, 65, 125
  pin type 23
primitive devices 11, 25, **73**
  delay 12
  input value mapping 15, 139
  pin order 143
  test vectors 68
Primitive Gates library 75
Primitive I/O library 26
Print Timing command 110
probe
  cursor 11, **126**
  device **92**, 146
Probe option 37
Programmable Logic Devices **99**
PROM device **99**
  editing 103
  size limits 100
  viewing contents 104
PromPLA tool 3, 101, **102**

pulldown resistor 78
pullup resistor 11, 15, **78**
Quine-McClusky method 100
RAM device 15, **95**, 126
  editing 98
  size limits 97
RAM tool 3, **96**
register device 14, 15, 19, **85**, 145
Remove command 34, 132
Report tool 26, 28
Reset option 48
resistive logic states 7, **11**, 78
resistor device 11, 15, **78**, 144
Resize Columns command 67
resolution 33
Restart option 37
retention options 130
ring oscillator 8
rise time 5
Run command 113
Save Test File As command 70
Save Test File command **57**
Save Test to Device command **68**, 139
Save Timing Text As command 110
Save to Attribute option 14, 116, **117**, 118, 119, 120
Save to Design Attr command 57
Select All command
  TestPanel 60
  Timing 39, **113**
Set from Attribute option 14, 116, 117, 118, 119, 120
Set Trace File command 57
setup and hold times **15**, 53, 70, 93, 118, 141
  flip-flops 83
SetupHold device 16, 83, **93**, 114, 137, 146
  setting 118
Shift key 26, 34, 38, 47, 58, 59, 92
shift register device 14, 15, **89**, 145
Show Timing Palette command 36, **131**
Signal Probe command **126**
signal probe tool 10, 11, 24, 37, **126**
signals
  0 and 1 24
  displaying in timing diag. 6, 128
  hierarchical names 131
  in triggers 122
  initial values 19, **20**
  logic states 6
  probing 126
  stuck value **9**, 124
Sim.InputMap attribute field 15, 139
Sim.TrigSave attribute field 139
SimLoad tool 3, 26, **27**
simulation **5**
  description 5
  speed 36, 113
Simulation menu 113
Simulation Params command 12, 13, **114**
  clock device 89, 90
  one shot 91
  One Shots 91
  SetupHold device 93
Simulation Scope command 125
simulation speed 36, 113
simulation time 13, 33, 36, 40, 41, 50
  triggers 122
Single Step command 37, **114**
SPDT switch device 15, **92**, 146
speed
  menu 113
  tool palette 36
Speed menu 113
SPST switch device 15, **91**, 146
Step option 37
Stick Signals command 10, **124**
Stop command 37, **113**
stuck signal values **9**
  0 and 1 signals 24
  clearing **10**, 124
  power and ground 24
  setting **10**, 63, 124
  signal probe 128
  Stick Signals command 124
  storage devices 14
  TestPanel 63
sub-circuits
  attributes 137
  checking external references 31
  creating external models 31

delay 12, 114, 115
external 25, **26**, 27, 138
input value mapping 15
internal 25
pin inversion 17
pin type 142
PLDs 99, 100
purging 30
reloading 30
signal names 131
SimLoad 27
simulation 25
TestPanel 45, 61, 65
updating 30
subtracter device **83**, 144
switch device
binary **91**, 146
SPDT 15, **92**, 146
SPST 15, **91**, 146
synchronizing
clocks 89
TestPanel 71
synchronous counter See counter device
Test Options command 52, 55, **62**, 71
test vectors **43**, 49
copying to timing diagram **66**
design 139
device 67, 139
example 69
TestPanel tool 3, **43**
clipboard data format 133
closing 47
commands 49
editing 47
examples 68
hierarchical designs 45
menus 57
multiple TestPanels 46
opening 46
TestVectors.Cct attribute field 139
TestVectors.Dev attribute field 68, 139
three-state buffer device **77**, 144
three-state outputs 7, 23, 141
PLDs 99
RAM 95
time
retention time 130
time units 6
tool palette display 36
time scale 33
Timing Data Retention option 130
timing diagram **33**
adding traces 34
clipboard data format 133
drawing tools 41
editing **37**, 41
reference lines 18, 123
removing traces 34, 132
resolution 33
status box 34
time scale 33
tool palette 34, 36, 131
Timing Options command **130**
timing text data format 133
Timing tool 3, **33**
Timing.Save attribute field 139
To Bottom command 34, 132
To Top command 34, 132
tool palette 131
controls 36
time display 36
timing 36
trademarks 147
transmission gate device **77**, 144
triggers 18, **121**
display on/off 37
setting **121**
setup and hold checking 83
Sim.TrigSave attribute field 139
Triggers command 37, **121**
Triggers option 37
unconnected inputs 7, 9, 19, 92, 126
Undo command 58, 110
Ungroup command 35, 41, **132**
unknown detector device 94, 146
Update Display command 67
X-Gate device **77**, 144
XNOR device **75**, 143
XOR device **75**, 143
zero delay **13**, 92, 106, 116
zoom in/out 37